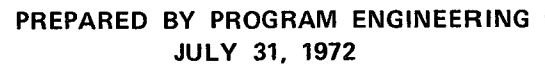


$$z^{(m)}(x)$$

SD 72-SA-0114-4

PHASE B EXTENSION

Volume IV: Data Processing Assembly



N72-33847

Unclas
16406



1 2 2 1 4 L a k e w o o d B o u l e v a r d D o w n e y , C a l i f o r n i a 9 0 2 4 1

TECHNICAL REPORT INDEX/ABSTRACT

ACCESSION NUMBER				DOCUMENT SECURITY CLASSIFICATION		UNCLASSIFIED	
TITLE OF DOCUMENT						LIBRARY USE ONLY	
INFORMATION MANAGEMENT ADVANCED DEVELOPMENT FINAL REPORT, Volume IV, Data Processing Assembly							
AUTHOR(S)							
Gerber, C. R., et.al.							
CODE	ORIGINATING AGENCY AND OTHER SOURCES				DOCUMENT NUMBER		
QN085282	Space Division of North American Rockwell Corporation, Downey, California				SD 72-SA-0114-4		
PUBLICATION DATE			CONTRACT NUMBER				
July 31, 1972			NAS9-9953				
DESCRIPTIVE TERMS							
<p>*MODULAR SPACE STATION, *INFORMATION MANAGEMENT, *ADVANCED DEVELOPMENT, *COMMUNICATIONS, *DATA PROCESSING, *SOFTWARE, *DATA BUS, *DATA STORAGE, *TECHNOLOGY, *SPECIFICATIONS</p>							

ABSTRACT

THIS DOCUMENT IS VOLUME IV OF THE FINAL REPORT OF THE MODULAR SPACE STATION ADVANCED DEVELOPMENT STUDY. THIS VOLUME (1) DEFINES THE COMPUTATION AND LOGICAL FUNCTIONS WHICH MUST BE PERFORMED BY THE DATA PROCESSING ASSEMBLY (DPA) FOR THE ORBITAL OPERATIONS, (2) DEFINES IN PRELIMINARY FORM THE MEMORY SIZE AND COMPUTER SPEED REQUIRED TO ACCOMPLISH THESE FUNCTIONS, (3) ALLOCATES COMPUTATIONS AND LOGICAL OPERATIONS TO ELEMENTS OF THE DPA, (4) DEVELOPS PRELIMINARY FLOW DIAGRAMS WHICH PORTRAY THE INFORMATION FLOW RATES AND FUNCTIONS PERFORMED BY THE DPA AND ITS INPUT/OUTPUT INTERFACE WITH THE MSS SUBSYSTEMS AND (5) DEFINES A DPA CONFIGURATION.

SD 72-SA-0114-4

MODULAR **space station**

PHASE B EXTENSION

INFORMATION MANAGEMENT ADVANCED DEVELOPMENT FINAL REPORT

Volume IV: Data Processing Assembly

31 JULY 1972
PREPARED BY PROGRAM ENGINEERING

Approved by



James Madewell
Director
Space Applications Programs



Space Division
North American Rockwell

FOREWORD

This document is one of a series required by Contract NAS9-9953, Exhibit C, Statement of Work for the Phase B Extension - Modular Space Station Program Definition. It has been prepared by the Space Division, North American Rockwell Corporation, and is submitted to the National Aeronautics and Space Administration's Manned Spacecraft Center, Houston, Texas, in accordance with the requirements of the Data Requirements List (DRL) MSC-T-575, Line Item 72.

This document is Volume IV of the Modular Space Station Information Management System Advanced Development Technology Report, which has been prepared in the following five volumes.

I	Summary	SD72- SA-0114-1
II	Communications Terminal Breadboard	SD72- SA-0114-2
III	Digital Data Bus Breadboard	SD72- SA-0114-3
IV	Data Processing Assembly	SD72- SA-0114-4
V	Software Assembly	SD72- SA-0114-5



ACKNOWLEDGEMENTS

The following persons have participated in the conduct of the IMS ADT tasks, and have contributed to this report:

C. W. Roberts	Experiment/Electronics Manager
C. R. Gerber	Information Systems Project Engineer
B. A. Logan, Jr.	Information Systems
E. Mehrbach	Information Systems
D. W. Brewer	Information Systems
V. R. Hodgson	Information Systems

The following subcontractors have supported the IMS ADT tasks in specialized areas:

International Tel. & Tel. Nutley, New Jersey B. Cooper, Proj. Mgr.	Communications Terminal B.B. Data Bus BB
Intermetrics Cambridge, Mass. J. Miller, Prog. Mgr.	Data Processing Assy
System Develop. Corp. Santa Monica, Calif. R. Bilek, Prog. Mgr.	Data Processing Assy Software Assy
Gen'l Electric Corp. Valley Forge, Pa. R. Kirby, Prog. Mgr.	Bulk Storage Technology
NR-Autonetics Anaheim, Calif. J. Jurison, Proj. Mgr.	Data Bus BB Data Processing Assy

Preceding page blank

PRECEDING PAGE BLANK NOT FILMED

CONTENTS

Section	Page
1. INTRODUCTION	1-1
1.1 APPROACH	1-4
1.2 DPA DEFINITION	1-15
1.3 DPA ENGINEERING MODEL DEVELOPMENT PLAN	1-19
2. REQUIREMENTS ANALYSIS	2-1
2.1 BASELINE DATA PROCESSING ASSEMBLY REQUIREMENTS	2-1
2.2 REFINEMENTS TO BASELINE REQUIREMENTS	2-7
3. BASELINE DPA CONFIGURATION	3-1
3.1 DISTRIBUTION OF SUBSYSTEMS REQUIREMENTS	3-1
3.2 CONFIGURATION TRADEOFFS	3-1
3.3 BASELINE DPA CHARACTERISTICS	3-11
4. INFORMATION FLOW STUDY	4-1
4.1 INTRODUCTION	4-2
4.2 DATA PROCESSING ASSEMBLY (DPA) CONFIGURATION	4-3
4.3 INFORMATION FLOW DIAGRAMS	4-3
5. THROUGHPUT SIMULATION	5-1
5.1 INTRODUCTION	5-1
5.2 TIME SLOT ANALYSIS	5-5
5.3 ANALYTICAL RESULTS	5-6
5.4 DPA SIMULATION MODEL	5-9
5.5 SIMULATION RESULTS	5-17
6. DPA REDUNDANCY STUDY	6-1
6.1 REDUNDANCY	6-1
6.2 DPA REDUNDANCY CONFIGURATION	6-1
7. CENTRAL PROCESSOR STUDIES	7-1
7.1 CP OPERATIONAL ANALYSES	7-1
7.2 HOLM MEMORY ORGANIZATION AND INTERNAL BUS DESIGN	7-25
7.3 REFERENCES	7-65
8. MEMORY STUDIES	8-1
8.1 MASS MEMORY TECHNOLOGY	8-1
8.2 ARCHIVAL MEMORY	8-5
9. DPA CONFIGURATION SELECTION	9-1
9.1 REVISIONS TO BASELINE DPA CONFIGURATION	9-1
9.2 CENTRAL PROCESSOR TASK ALLOCATIONS	9-8

Preceding page blank

Section		Page
10.	PROCESSORS PERFORMANCE REQUIREMENTS	10-1
10.1	CENTRAL PROCESSOR TYPE	10-1
10.2	CP INTERNAL STRUCTURE	10-2
10.3	HOLM VERSUS CONVENTIONAL ARCHITECTURE	10-4
10.4	SUMMARY OF PROCESSORS CHARACTERISTICS.	10-9
11.	DPA DESIGN REQUIREMENTS SPECIFICATION	11-1
11.1	DATA PROCESSING ASSEMBLY GENERAL SPECIFICATIONS	11-1
11.2	MULTIPROCESSOR SET SPECIFICATION	11-12
11.3	MASS MEMORY SPECIFICATION	11-23
11.4	ARCHIVE MEMORY SPECIFICATION	11-25
11.5	PREPROCESSOR SPECIFICATIONS	11-27
12.	DMS PROCESSOR EEM DEVELOPMENT PLAN	12-1
12.1	OBJECTIVE	12-1
12.2	PROGRAM PLAN	12-2
12.3	EEM PROCESSOR PLAN	12-7
12.4	SCHEDULING OPTIONS	12-15

ILLUSTRATIONS

Figure		Page
1-1	DPA General Diagram	1-2
1-2	Data Processing Assembly Distribution	1-3
1-3	Data Processing Assembly Configuration Study	1-5
1-4	Processor Utilization	1-8
1-5	Basic Recommended Redundancy Configuration	1-9
1-6	Simplex Multiprocessor Schematic	1-11
1-7	Memory Hierarchy	1-12
1-8	Internal Bus Configurations	1-13
1-9	Detailed CP Subsystem Functional Allocations	1-16
2-1	Requirements Analysis Study Approach	2-2
2-2	Phase B Requirements Computation Rationale	2-8
3-1	Baseline DPA Block Diagram	3-12
4-1	Power Module Hookup - Baseline	4-4
4-2	Core Module Hookup - Baseline	4-5
4-3	Station Module No. 1 Hookup - Baseline	4-6
4-4	Station Module No. 2 and No. 3 Hookup - Baseline	4-7
4-5	Station Module No. 4 Hookup - Baseline	4-8
4-6	Cargo Modules No. 1 and No. 2 Hookup - Baseline	4-9
4-7	G&C Information Flow Diagram	4-11
4-8	EPS Information Flow Diagram	4-13
4-9	ETC/LSS Information Flow Diagram	4-15
4-10	RCS Information Flow Diagram	4-17
4-11	Crew Subsystem Information Flow Diagram	4-19
4-12	Structures Subsystem Information Flow Diagram	4-21
4-13	Communications Assembly Information Flow Diagram	4-23
4-14	ISS Information Flow Diagram - Central Processor	4-25
4-15	Information Flow Example (Antenna Pointing) Central Processor Level	4-29
4-16	Information Flow Example (Antenna Pointing) Subsystem Level	4-31
5-1	IMSIM Workload Hierarchy	5-10
5-2	IMSIM Implementation of CP Configuration	5-16
5-3	Processor Utilization	5-24
6-1	Basic Recommended Redundancy Configuration	6-3
6-2	Data Transmission Error Checkers	6-7
6-3	Short Message Echo Checker	6-7
6-4	State Diagram for Fault Transitions	6-14



Figure		Page
7-1	Simplex Multiprocessor Schematic	7-8
7-2	Block Code for M2	7-18
7-3	Language and Machine	7-26
7-4	Memory Hierarchy	7-31
7-5	M1 Functions	7-33
7-6	5-Way Interleaved M2 Complex	7-39
7-7	Internal Bus	7-42
7-8	Internal Bus Configurations	7-43
7-9	Internal Bus Delay Factors	7-49
7-10	Fault Tolerant Multiprocessor.	7-56
9-1	Buildup Sequence - Initial MSS Complexes	9-2
9-2	DPA General Diagram	9-3
9-3	Revised DPA Diagram	9-4
9-4	Detailed CP Subsystem Functional Allocations	9-10
10-1	CP Internal Structure.	10-3
11-1	Data Processing Assembly	11-3
11-2	CP Block Diagram	11-10
11-3	Central Processor Block Diagram	11-13
11-4	Preprocessor Block Diagram	11-28
12-1	EEM Processor Program Schedule	12-3
12-2	ADT Breadboards Related to ADT Plan	12-4
12-3	Relationship of EEM Processor Development to ADT Extension	12-5
12-4	EEM Processor Development Schedule	12-9
12-5	Procurement Options for a Dual Multiprocessor EEM.	12-17

TABLES

Table	Page
1-1 Computation Requirements for Station Operations . . .	1-6
1-2 Redundancy Recommendations	1-7
1-3 Summary of Memory Hierarchy	1-10
1-4 Internal Bus Major Conclusions	1-14
1-5 Fault Tolerance Recommendations	1-14
1-6 Technical Characteristics of the Central Processor . .	1-17
1-7 Technical Characteristics of the Preprocessor . . .	1-18
2-1 DPA Interface Requirements	2-5
2-2 Computation Requirements for Station Operations . . .	2-9
3-1 DPA Alternatives for Modular Space Station	3-2
3-2 MSS Signal Allocation Assumptions	3-3
3-3 Central Processor Selection	3-10
3-4 Summary of Computer Sizing	3-14
4-1 Preliminary DPA Interconnection Notes	4-10
5-1 Arithmetic Unit Utilization Analysis.	5-8
5-2 Workload Summary	5-19
7-1 Requirements and Data Types	7-4
7-2 Internal Bus Traffic Due to Instruction Execution . .	7-45
7-3 Bus Traffic Summary for Configurations	7-47
7-4 Comparison of Number of Drivers and Receivers . . .	7-51
7-5 Internal Bus Evaluation Summary	7-51
7-6 DPA Failure and Error Tolerance Criteria	7-53
8-1 Archival Storage Tradeoff Table	8-7
9-1 Central Processor Computational Requirements	9-6
9-2 RPU Sizing	9-6
9-3 RACU Utilization	9-7
9-4 Processor Allocations	9-12
9-5 AU-IO ₁ Subsystem Distribution	9-14
9-6 AU-IO ₂ Subsystem Distribution	9-14



Table		Page
10-1	CP Hardware Comparison	10-1
10-2	CP Memory Requirements	10-5
10-3	HOLM Implementation	10-6
10-4	Cost Evaluation of HOLM and Conventional Design	
	Applied to MSS	10-7
10-5	Cost Savings Between HOLM and Conventional Organizations	10-8
10-6	Summary of HOLM-Conventional Organization Evaluation	10-8
10-7	Technical Characteristics of the Central Processor	10-10
10-8	Technical Characteristics of the Preprocessor	10-11
11-1	6-Man MSS Hardware Location	11-4
11-2	Electrical Interfaces	11-6
11-3	Multiprocessor Electrical Interfaces	11-16
11-4	Preprocessor Performance Characteristics	11-30
11-5	Input-Output Performance Characteristics	11-31
12-1	Test Coverage of Key Features with Various Breadboard Processors	12-3



ABBREVIATIONS

ADS	Advanced Data System
ADT	Advanced Development Technology
ADTX	Advanced Development Technology Extension
AFC	Automatic Frequency Control
AGC	Automatic Gain Control
AN	Autonetics (Division of North American Rockwell Corporation)
BPF	Bandpass Filter
bps	Bits Per Second
CAIRS	Computer-Assisted Interactive Resource Scheduling
CCIR	International Radio Consultative Committee
CDR	Critical Design Review
CEI	Contract End Item
CLASP	Computer Language for Aeronautics and Space Programming
CM	Command Module (Apollo)
COMPOOL	Common Pool (of Data)
CP	Central Processor or Circular Polarization
C.P.	Computer Program
CPCEI	Computer Program Contract End Item
CPCI	Computer Program Configuration Item
CPDF	Computer Program Development Facility
CPIC (A)	Computer Program Integration Contractor (Agency)
CPT&E	Computer Programming Test and Evaluation
CR	Change Report
CRT	Cathode-Ray Tube (Display)
CSS	Crew Subsystem
CTB	Communications Terminal Breadboard
CTF	Central Test Facility
DACS	Data Acquisition and Control Subassembly
dB	Decibel
DBC	Data Bus Control Unit
dBm	Decibel Referred to One Milli-Watt
dBW	Decibel Referred to One Watt
DCR	Design Change Request
DDB	Digital Data Bus
Demux	De-Multiplex(er)
DMS	Data Management System
DPA	Data Processing Assembly
DPSK	Dual Phase Shift Keying
DRSS	Data Relay Satellite System



ECP	Engineering Change Proposal
EDF	Experiment Data Facility
EEM	Engineering Evaluation Model
EEMP	Engineering Evaluation Model Processor
EIRP	Effective Isotropic Radiated Power
EMC	Electromagnetic Compatability
EMI	Electromagnetic Interference
EOS	Earth Orbital Shuttle
EOSS	Earth Orbital Space Station
EPS	Electrical Power Subsystem
ETC/LSS or ECLSS	Environment Control and Life Support Subsystem
EVA	Extra-Vehicular Activity
EXT	External
E _b /N ₀	Energy Per Bit to Noise Density Ratio
FACS	Facsimile
FDM	Frequency-Division Multiplex
FM	Frequency Modulation
FQT	Formal Qualification Test
G&CS	Guidance and Control Subsystem
GFE	Government Furnished Equipment
GHz	Giga-Hertz
GOA	Gated Operational Amplifier
HAL	Higher-Order Aerospace Programming Language
HOL	Higher-Order Language
HOLM	Higher-Order Language Machine
Hz	Hertz
IF	Intermediate Frequency
IFRU	In-Flight Replaceable Unit
IM	Intermodulation Products
IMS	Information Management System
IMSIM	Information Management Simulation
IOC	Initial Operational Capability
IOCB	Input-Output Control Block
IOU	Input-Output Unit
I/O	Input-Output
IPA	Intermediate Power Amplifier
IQL	Interactive Query Language
IR	Infra-Red
ISS or IMS/S	Information (Management) Subsystem
ITT	International Telephone and Telegraph
K-words	Thousands of (Computer) Words
K-EAPS	Thousands of Equivalent-Add Operations Per Second
K-bps	Thousands of Bits Per Second
KHz	Kilohertz

LEM	Lunar Excursion Module
LM	Lunar Module
LNA	Low Noise Amplifier
LO	Local Oscillator
LPF	Low Pass Filter
M1, M2	(Computer) Memory Designation
Mbps	Megabits Per Second
MCB	Module Control Block
MHz	Megahertz
MOF	Mission Operations Facility
MOL	Manned Orbiting Laboratory
MSC	Manned Spacecraft Center
MSFN	Manned Space Flight Network
MSS	Modular Space Station
MUX	Multiplexer
mW	Milli-Watts
MW	Microwave
mV	Milli-Volts
NF	Noise Figure
OBCO	On-Board Checkout
OCC	Operations Control Center (On-Board)
ODM	Operational Data Management
OM	Operating Memory
PA	Power Amplifier
PCM	Pulse Code Modulation
PDR	Preliminary Design Review
PL/1	Procedure Language
PM	Phase Modulation
PN (PRN)	Pseudo Random Noise
ppm	Parts Per Million
PQT	Preliminary Qualification Tests
PSK	Phase Shift Keying
RAM	Research and Applications Module
RACU	Remote Acquisition and Control Unit
RCS	Reaction Control Subsystem
RF	Radio Frequency
RHCP	Right-Hand Circular Polarization
RPU	Remote Processing Unit
Rx	Receive
S&C	Standards and Conventions
SCCB	Software Configuration Control Board
SCN	Specification Change Notice
SD	Space Division (of North American Rockwell Corporation)
SDC	Systems Development Corporation



S/N	Signal to Noise Ratio
SOW	Statement of Work
SPL	Space Programming Language
SRD	Step-Recovery Diode
SSCB	Solid-State Circuit-Breaker
SSS	Structures Subsystem
STE	Support Test Equipment
TAV	Test and Validation (Programs)
TBD	To Be Determined
TCXO	Temperature-Controlled Crystal Oscillator
TDA	Tunnel Diode Amplifier
TDM	Time Division Multiplexing
TDRS	Tracking and Data Relay Satellite
TIP	Test and Integration Plan
TLM, TM	Telemetry
TOOL	Test Operations Oriented Language
TRW	Thompson Ramo Woolridge Corporation
TT&C	Telemetry, Tracking and Control
TWT	Traveling Wave Tube
TWTA	Traveling Wave Tube Amplifier
Tx	Transmit
USB (E)	Unified S-Band (Equipment)
UV	Ultra-Violet
VDD	Version Description Document
VHF	Very High Frequency
VSF	Vestigial Side Band
VSWR	Voltage Standing Wave Ratio

LIST OF INTERIM REPORTS

AA-101	DPA Flow Diagrams, September 1971
AA-102	DPA Throughput and Authority Analysis, February 1972
AA-103	DPA Configuration Selection, April 1972
AS-101	Modular Space Station Computer Program Standards and Conventions, December 1971
AS-102	Modular Space Station Computer Program Specification Tree, February 1972
AS-103	Modular Space Station Computer Program Development, Test and Configuration Control Plan, May 1972
AS-104	Modular Space Station Computer-Assisted Resource Allocations and Utilization Reccomendations, June 1972
CTB-101	Concepts for Multiple RF Link Mechanization, May 1971
CTB-103	Antenna-Mounted Electronics Component Design, October 1971
CTB-105/106	CTB Integration and Test and Operations Manual, June 1972
DB-101	Parametric Data for Bus Design, May 1971
DB-103	Component Performance Requirements, Schematics and Layout Drawings, December 1971
DB-104	Digital Data Bus Breadboard Final Report, May 1972
DD-102	Modular Space Station Data Processing Assembly Parametric Evaluation of Subsystems Input/Output Interface, June 1971

DD-103	Modular Space Station Data Acquisition and Control Subassembly Model Configuration (SD 71-233), July 1971
DP-101	Data Processing Assembly Configuration (Preliminary), June 1971
DP-102	Data Processing Assembly Supervisor Specification, May 1972
DP-103	DPA Processor Performance Requirements (Preliminary), August 1971
DP-103	DPA Processor Final Description, May 1972
DP-104	EEM DMS Processor Development Plan, June 1972
DP-105	Data Acquisition and Control Redundancy Concepts, August 1971
DP-106	Application of Redundancy Concepts to DPA, January 1971
DP-107	Data Acquisition and Control Subassembly Breadboard Design Requirements, October 1971
DP-108	Data Bus Control Unit Performance Requirements, January 1972
DP-109	Data Bus Control Unit Design Reports, March 1971
DP-110	DBCUC Acceptance Report (to be published)
EL-277	Bulk Storage Development Plan
IB-101	DPA Internal Flow and Traffic Pattern, May 28, 1971
ICD #TRW 20549	Interface Control Document - Data Bus Modem/RACU, Revision A, January 17, 1972
ICD #AN 26465	Interface Control Document - Data Bus Controller Unit to Buffer I/O, Revision January 21, 1972
MD-101	Mass Memory Parametric Data
RF-101	Modular Space Station Communications Terminal Breadboard Preliminary System Specification, October 1971

SA-101	Central Processor Operational Analysis, September 30, 1971
SA-102	Central Processor Memory Organization and Internal Bus Design, December 30, 1971
SD 71-227	Automatic Control and Onboard Checkout Final Study Report

1. INTRODUCTION AND TASK SUMMARY

The requirements for a long duration manned space station include continuous maintenance of operational capability with minimum crew participation. This requirement can be achieved by automating operations of the subsystem functions with use of a computer system, hereafter referred to as the Data Processing Assembly (DPA).

Volume IV summarizes the efforts directed to defining the DPA data input/output requirements and traffic flow patterns, allocating logical and computational functions for the development of information flow diagrams and defining a DPA configuration.

The computations required may be performed in a number of ways. The concepts, performance, mechanizations, reliability and cost are sensitive to the amount of automation required. The approach taken was to (a) define the computation and logical functions which must be performed by the data processing assembly (DPA) for the modular space station (MSS) orbital operations, (b) define in preliminary form the memory size and computer speed required to accomplish these functions, (c) allocate computations and logical operations to elements of the DPA, (d) develop preliminary flow diagrams which portray the information flow rates and functions performed by the DPA and its input/output interface with the MSS subsystems and (e) define a DPA configuration.

Figures 1-1 and 1-2 present the configuration selected for the Data Processing Assembly. As noted the station operations Central Processor (CP) is located in the primary Control Module (SM-1). Supervisory control of the equipment in the Power and Core Modules is provided via a radio link during station buildup prior to SM-1 arrival. A special component (Build-up Data Processor) is located in the Core Module for interfacing with the radio link and DPA. This component will be removed or disengaged when SM-1 arrives and supervisory control will then be exercised by the station operations Control Processor. The baseline configuration is further shown to consist of remote processing units (RPU's) performing certain subsystem functions (particularly G&C) and failure detection. A redundant bus network connects these and the Remote Access Control Units (RACUs) to the central processor.

A multiprocessor organization has been defined as the most suitable for the central processor. Redundancy at the central control level is further supplied by another central computer containing the critical operations functions and experiment support software. This second Central Processor is located in another pressure volume (SM-4) and is identical to the primary computer. The RPU's consist of uniprocessors with special input/output processing or signal processing as required to accommodate the subsystem functional requirements.

* Not part of DPA

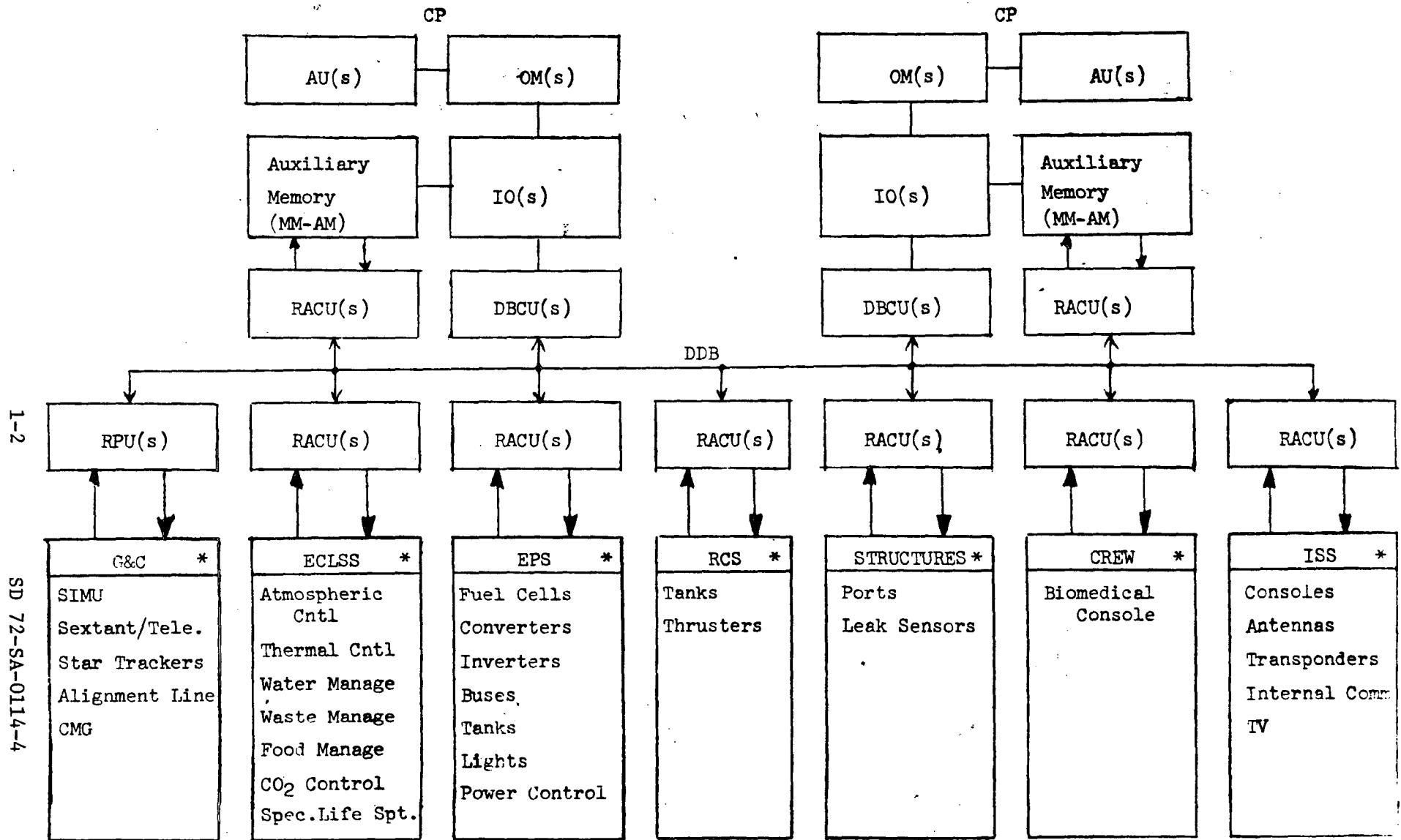


Figure 1-1. DPA General Diagram

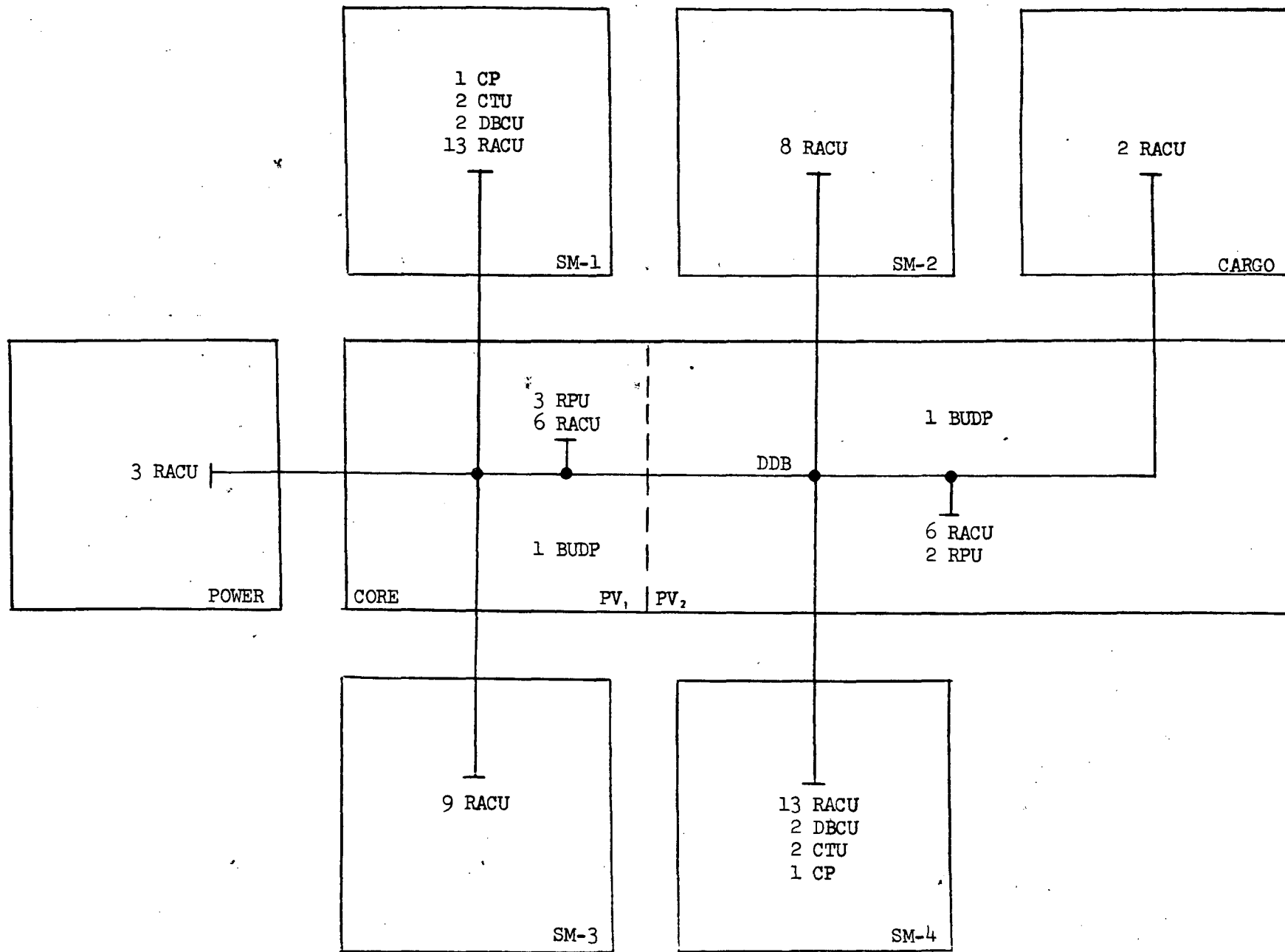


Figure 1-2. Data Processing Assembly Distribution

1.1 APPROACH

Figure 1-3 shows the tasks, their relationships, and the subcontractors who participated in each task. The impact of the simultaneous MSS Phase B study is also indicated. It will be noticed in reading this report that many different values of memory size and operating speed are used. Basically, this is due to two factors: the DPA studies were impacted by the Phase B studies and the DPA studies were iterative (particularly as regards the selection of a DPA configuration). The most significant difference between two sets of DPA requirements is due to the ongoing requirements and subsystems analysis in the Phase B study. Once past the insertion of this large delta, the differences in assumed requirements is minor (not more than 10%) and does not significantly alter the DPA concept or the results of the study.

The study began with an analysis of the DPA requirements. This task defined the subsystems' functions which require data processing support; defined the mechanization required to provide data processing support for each identified function; estimated the memory, speed and input/output data rates required for mechanization of each function; and integrated the subsystems' computation requirements to define a total set of MSS DPA requirements. As indicated by Figure 1-3, the requirements analysis was continued throughout the Phase B effort, and finally resulted in a significant reduction of the DPA requirements. Table 1-1 presents the resulting performance requirements for station operations in parameters of processing speed, memory capacity and data bus rate. Shown are the basic requirements, the design margin, the growth margin, the initial design requirements and the maximum design requirements.

The next task was the definition of the baseline DPA configuration. The alternatives to be considered at each processing level were enumerated and the distribution of the signals (subsystem interfaces) was tabulated based on the physical distribution of the MSS subsystems. A tradeoff was performed which resulted in the selection of a central multiprocessor plus subsystem preprocessing as required. The multiprocessor-to-subsystem interfaces are to be implemented with Remote Acquisition and Control Units (RACUs) which communicate with the central processor via a digital, time-serial data bus.

The purpose of the information flow study was to define the MSS DPA information flow so that the DPA could be simulated using NASA's IMSIM (a simulation model used to assess various computer configurations). A method of flow diagram presentation and attendant tabulations was carefully selected to provide a comprehensive data file of software and information characteristics that will prove beneficial in the continuation of the Advanced Development Tasks and related studies. This data file consists of descriptions of each subsystem, baseline configuration data, buildup information, DPA computational loads and allocations, computer sizing information, message tabulation, signal interface lists, and DPA parametric data requirements.

The objective of the DPA throughput simulation was to provide information that would facilitate a selection of the final DPA configuration for the MSS; in particular, to provide information pertaining to DPA component performance that would yield a DPA configuration capable of accommodating imposed workloads within required response times.

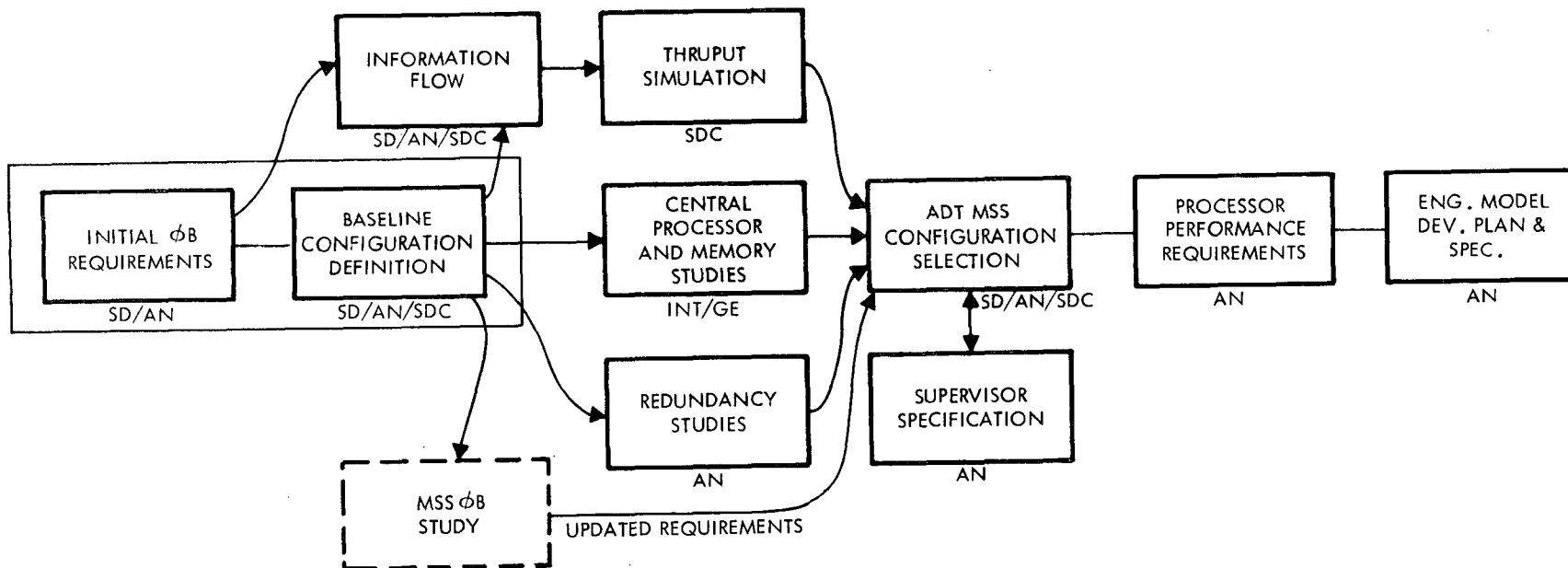


Figure 1-3. Data Processing Assembly Configuration Study

Table 1-1. Computation Requirements for Station Operations

	Performance Requirements	Base Requirements		Maximum Rqmts.		Initial (6-Man) Rqmts.	
		Basic Rqmts.	Design Margin (100%)	Maximum Growth Margin (100%)	Maximum Design Rqmts.	Initial Growth Margin	Initial Design Rqmts.
Central Processor	Processing Speed (Equiv. Adds/Sec)	631K	631K	631K	1893K	0	1262K
	Operating Memory (32 Bit Words)	67K	67K	67K	201K	0	134K
	Mass Memory (32 Bit Words)	341K	341K	341K	1023K	0	682K
	Data Bus Rate (Bits/Sec)	400K (Station Opn)	400K 2000K (experiments)	7.2M*	10M	7.2M	10M
	Archive Memory (32 Bit Words)	4.2M	4.2M	4.2M	12.6M	0	8.4M
Remote Processing Unit	Processing Speed (Equiv Adds/Sec)	125K	125K	0**	250K	0	250K
	Memory (32 Bit Words)	9K	9K	0**	18K	0	18K

* = Special Allowance for Experiments.

** = RPU Growth will be Accommodated by Additional Units.



The operational doctrine assumed to be in effect for the data bus is that of polling. Polling control is assumed to be a function of the I/O unit and the polling schedule is assumed to be on a "fixed" time basis per device. The polling schedule assumed is predicated on dividing a second into 250 slots of 4 ms each.

Nine time slots (36 ms) were simulated; these were the initial nine slots of each one second interval and were chosen for imposing the largest operational load on the DPA. Processor utilization during the execution of the simulation is shown in Figure 1-4. The simulation led to the following conclusions: the central processor can effectively process expected workloads, arithmetic unit speed of 750 KEAPs is adequate, the I/O processor is under utilized at 750 KEAPs, a data bus commutation cycle of 250 slots per second is reasonable, the operating memory transfer rate of 2×10^6 words per second is adequate and the mass memory transfer rate of 1×10^6 words per second is adequate. Note - as indicated by Figure 1-3, this task was conducted using the early Phase B requirements in contrast to the later Phase B requirements which were finally adopted.

The application of redundancy to the DPA stems from the failure criteria established for the MSS. The redundancy study was directed toward applying the criteria to the DPA concepts and recommending a satisfactory operational system. The recommended redundancy configuration is shown in Figure 1-5. The redundancy recommendations are shown in Table 1-2.

Table 1-2. REDUNDANCY RECOMMENDATIONS

CENTRAL PROCESSOR	TWO CP COMPLEXES
	EACH COMPLEX CAN TOLERATE ONE FAILURE AND DETECT ANOTHER FAILURE
	EACH COMPLEX CAN PROVIDE BACKUP OF CRITICAL FUNCTIONS
DATA BUS	TWO PLUS TWO ORGANIZATION
	ERROR PROTECTION CODE FOR ERROR DETECTION
	RETRANSMISSION FOR ERROR CORRECTION
DBCW	CONTROLS ALL FOUR BUSES
RACU	DUAL BUS INTERFACE
	SIMPLEX SUBSYSTEM INTERFACE

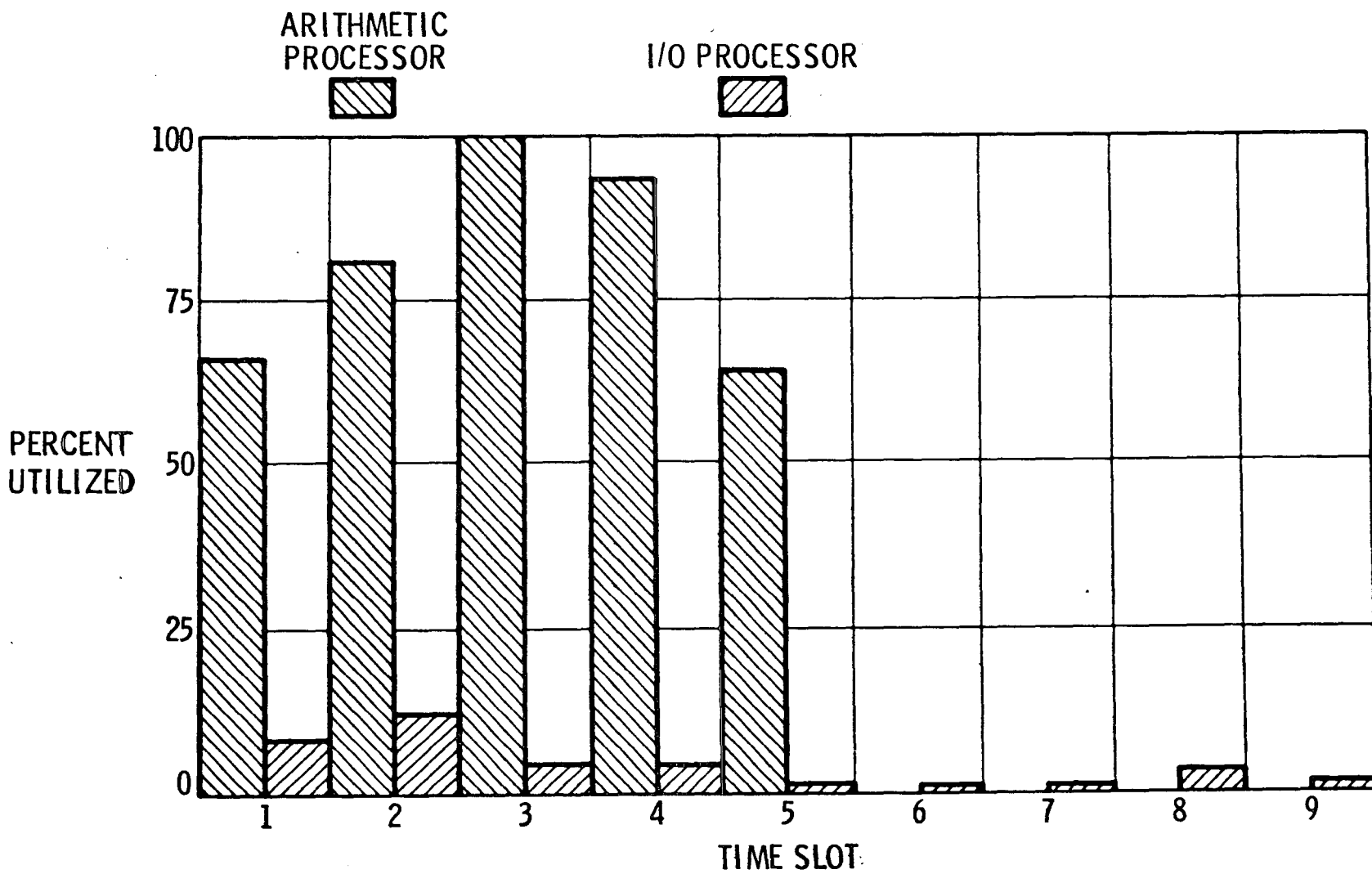


Figure 1-4. Processor Utilization

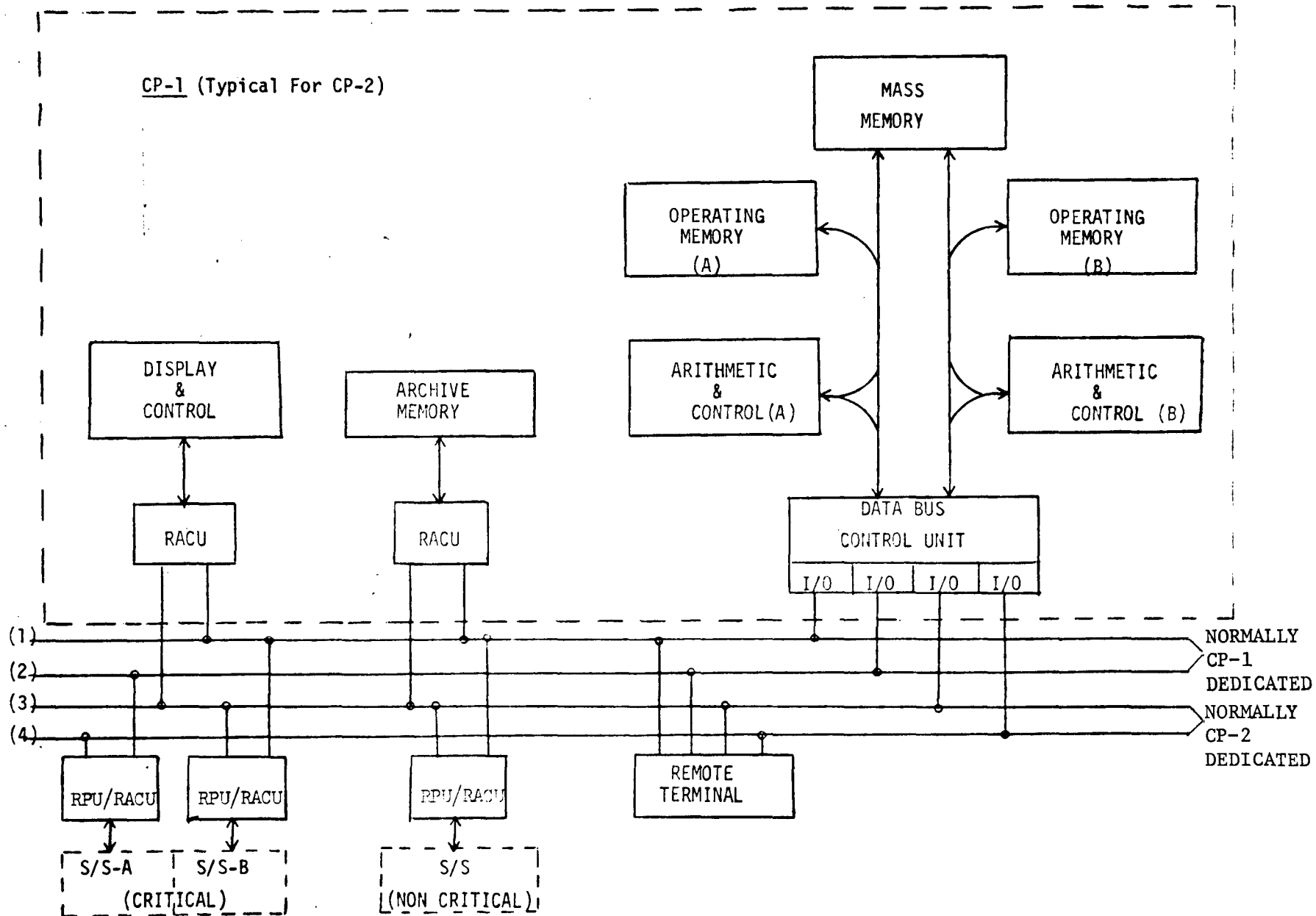
CHANNEL
NUMBERS

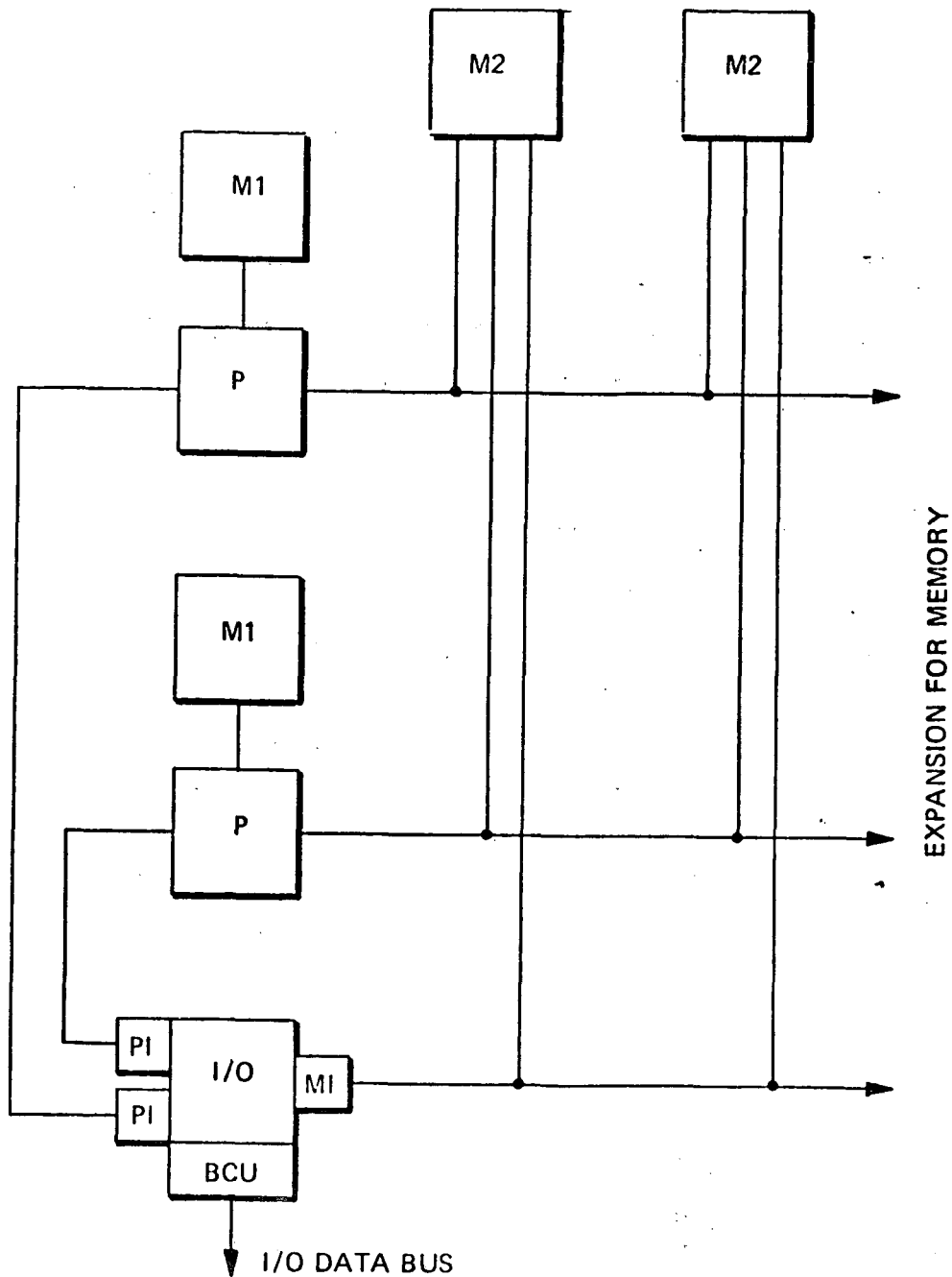
Figure 1-5. Basic Recommended Redundancy Configuration



The objective of the central processor study was to determine the MSS central processor operational use and software organization on the design of the hardware aspects of the central processor. The architecture of the central processor was recommended to be as shown in Figure 1-6. The concept of a Higher Order Language Machine (HOLM) was studied as a basis for studying the central processor memory hierarchy and the internal bus design. Figure 1-7 shows the memory hierarchy which was studied for the MSS Data Processing Assembly; Figure 1-8 presents the candidate internal bus configurations. Tables 1-3 and 1-4 summarize the conclusions reached during the study. Further analyses of fault tolerance for the central processor were conducted for the HOLM. The conclusions of those analyses are presented in Table 1-5.

Table 1-3. SUMMARY OF MEMORY HIERARCHY

<u>M1-LOCAL MEMORY</u>	
FUNCTION	STACKS, DESCRIPTORS, INSTRUCTION BUFFER, DATA VALUES
SPEED	200 - 500 ns
SIZE	400 - 32 BIT WORDS
TECHNOLOGY	CMOS LSI
<u>M2-OPERATING MEMORY</u>	
FUNCTION	CRITICAL INSTRUCTIONS, REDUNDANT CRITICAL DATA, OVERLAY AREA
SPEED	FIVE-WAY INTERLEAVED 1 MICROSECOND MEMORY MODULES PER M2 COMPLEX. DATA RATE 160 MBPS.
SIZE	TWO M2 COMPLEXES. EACH COMPLEX 80K 32 BIT WORDS
TECHNOLOGY	PLATED WIRE
<u>M3-MASS MEMORY</u>	
FUNCTION	NON-CRITICAL INSTRUCTIONS AND DATA REDUNDANT COPIES OF CRITICAL PROGRAMS
SPEED	FIVE MILLISECONDS LATENCY 6 - 10 MBPS TRANSFER RATE
SIZE	10 ⁶ 32 BIT WORDS
TECHNOLOGY	DRUM - LOW RISK PLATED WIRE - HIGH RISK



BCU - BUS CONTROL UNIT FOR DATA BUS
 MI - I/O MEMORY INTERFACE
 PI - PROCESSOR I/O INTERFACE

Figure 1-6. Simplex Multiprocessor Schematic

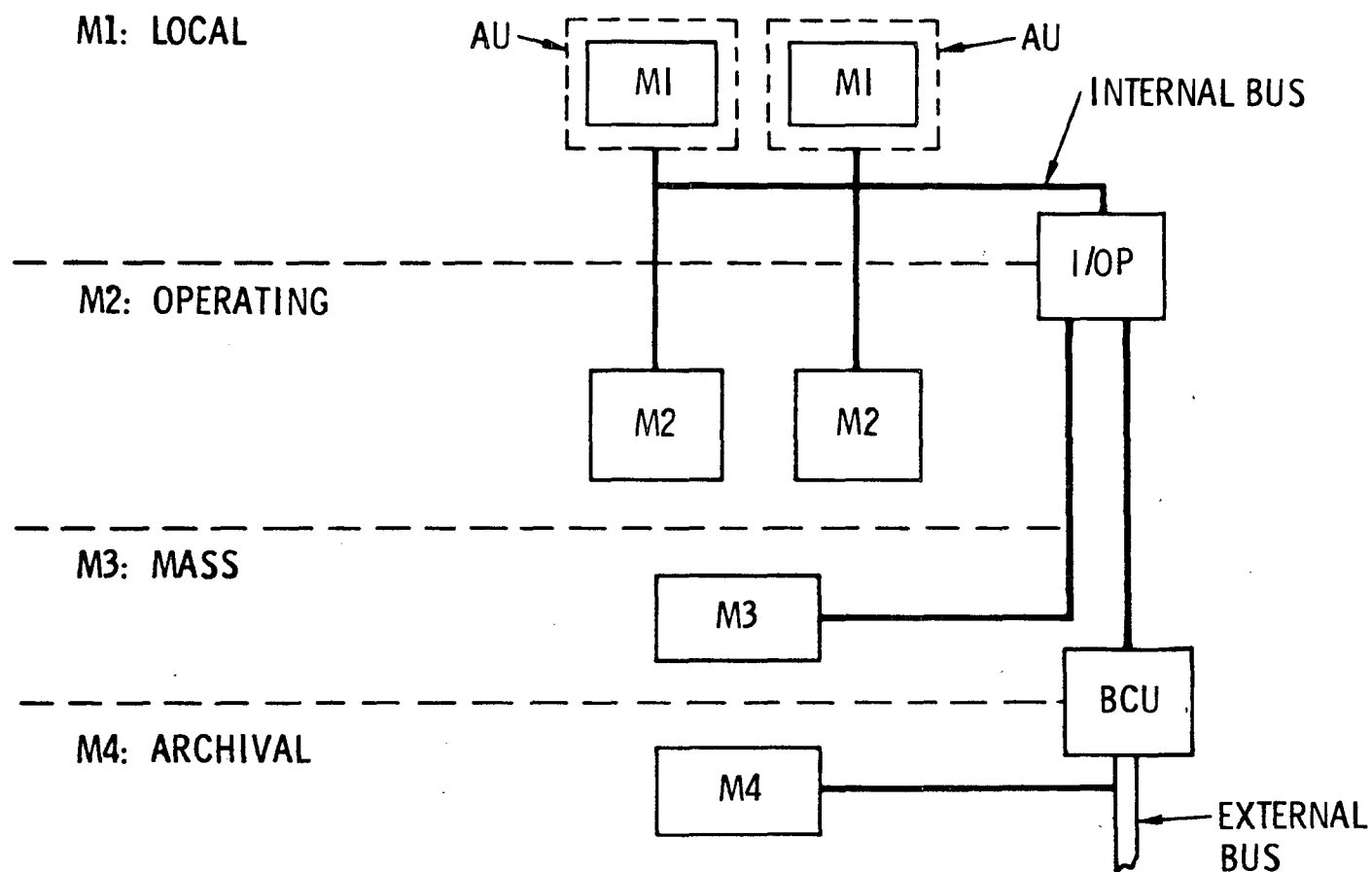
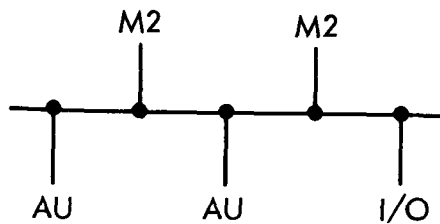
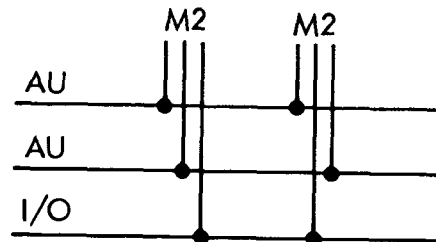


Figure 1-7. Memory Hierarchy

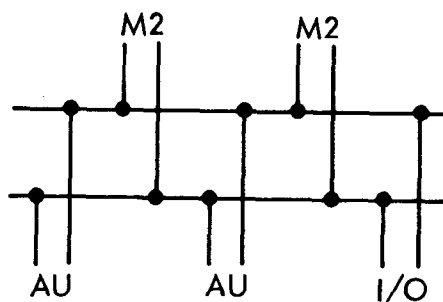
CONFIGURATION 1
 SINGLE TIME MULTIPLEXED BUS



CONFIGURATION 3
 DEDICATED BUSES WITH
 MULTIPOINT M2



CONFIGURATION 2
 MULTIPLE TIME SHARED BUSES
 WITH MULTIPOINT ELEMENTS



LEGEND:


 PHYSICAL
 CONNECTION

Figure 1-8. Internal Bus Configurations

Table 1-4. INTERNAL BUS MAJOR CONCLUSIONS

CHARACTERIZATION	Internal Bus B in M2 switching network and should be packaged with M2
STRUCTURE	Dedicated
WIDTH	32 bits
TRANSFER RATE	5 MBPS maximum
MODE	Synchronized interfaces

Table 1.5 FAULT TOLERANCE RECOMMENDATIONS

AU-MI	INTERNAL BUS
DUAL REDUNDANT AU'S WITH COMPARATOR FOR ERROR DETECTION	NOT REDUNDANT WORD PARITY ERROR DETECTION
GENERATES M2 PARITY AND CHECKWORD	M3 RECOVERY NOT REQUIRED
INDICATES M2 ERRORS BY WRITE ECHO CHECK	FAILURE DETECTION SIMILAR TO M2
RECONFIGURATION VIA SOFTWARE RESTART AT LEVEL OF SCHEDULED TASK	
M2	I/O
TWO INDEPENDENT M2 COMPLEXES CONTAINING CRITICAL PROGRAMS, REDUNDANT CRITICAL DATA, AND OVERLAY AREA	TRIPLE REDUNDANCY WITH VOTERS M2 FAILURE DETECTION AS PER AU
BLOCK PARITY WITH IMBEDDED ADDRESS FOR ERROR DETECTION	DBCUC GENERATES F.S. COMMUNICATION
SOFTWARE RECONFIGURATION THRU REALLOCATION OF M2 SPACE TO CRITICAL TASKS	SUBSYSTEMS PERFORM THEIR OWN WIND DOWN IN A F.S. SITUATION

Subsequent to the selection of a baseline DPA configuration, several of the influencing factors were changed as a result of the concurrent MSS Phase B definition studies. Most notable of these factors were the new buildup sequence for the MSS, the redefined DPA failure and error tolerance criteria, and the redefined computational requirements. Based on these new factors and the studies that were completed, the DPA configuration was redefined as shown, in Figures 1-1 and 1-2. An analysis was also performed to determine the effects of a more efficient distribution of the processing tasks within the central processor between the arithmetic units and the input/output processors. Figure 1-9 presents a detailed allocation of the central processor functions which resulted from the analysis.

1.2 DPA DEFINITION

The Data Processing Assembly (DPA) provides the computing functions needed for a high degree of reliable automation in the Modular Space Station. The central processors and preprocessors are key elements of the DPA and must perform reliably and effectively over the life of the station. The processor performance requirements task covers the central processor and preprocessor in terms of their internal organization and required functional and performance characteristics.

The central processor is a multiprocessor which possesses the features shown in Table 1-6. As noted, a conventional organization is preferred. A memory hierarchy consisting of buffer memories in the processing elements, modular operating and mass memories is provided. The requirements can be met with two arithmetic and input output processing sets. Each set contains dual units with capability of comparing memory addressing, controls, and processed results.

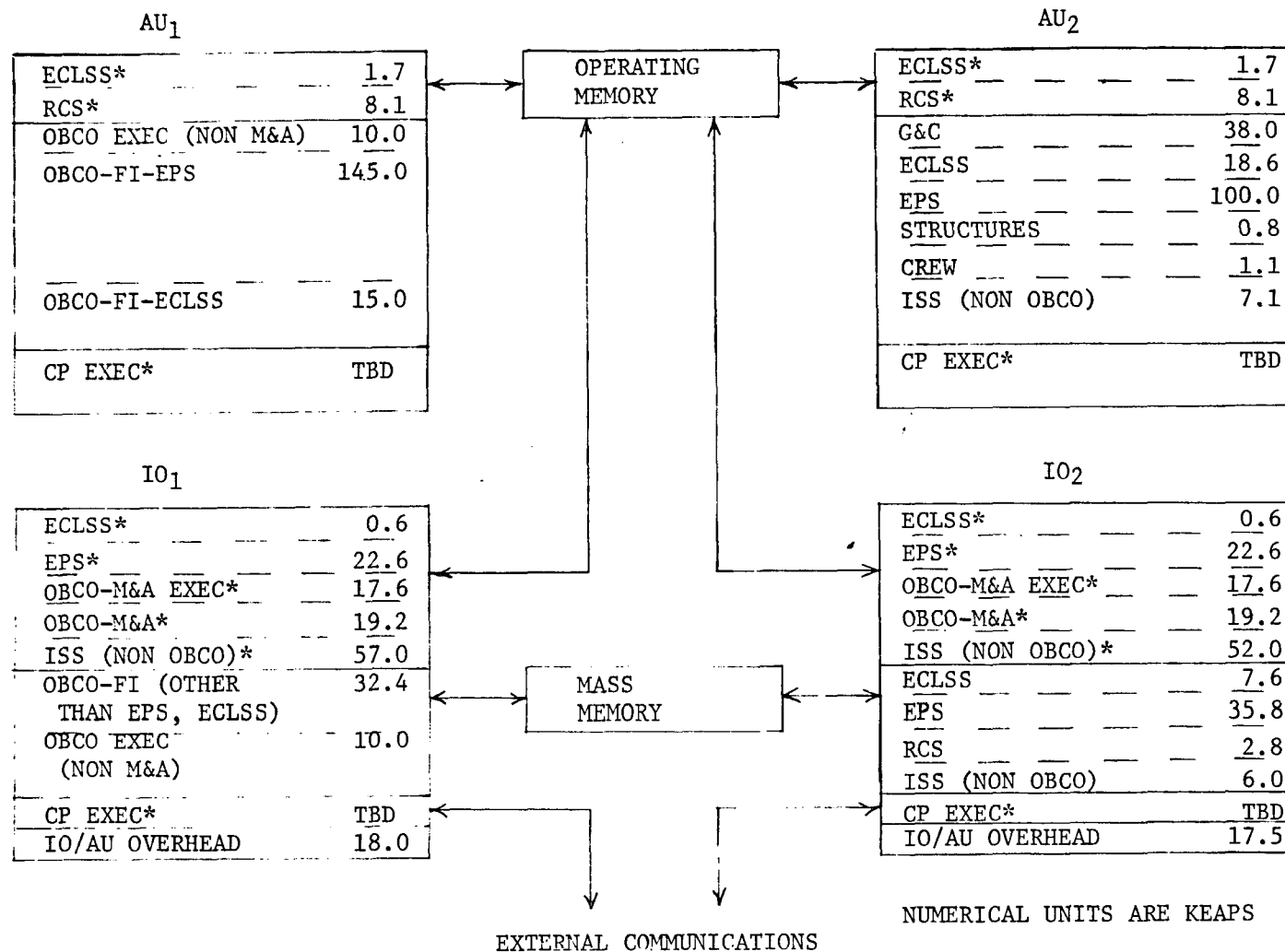
The central processor utilizes two operating memories for the main storage functions. These memories are supplied by paging techniques with information from a mass memory. Additional offline storage is provided by an archive memory. The key features of these are tabulated in Table 1-6.

An arithmetic unit provides one million equivalent adds per second capability. An extensive repertoire, including floating point, is incorporated into the design. Modes of operation include the normal computational and the executive. Privileged instructions only executable in the latter are used. Linkage to the executive mode is by interrupts and special instructions.

All input/output functions are controlled by the AU's by means of I/O control words and commands from the AU's. Once initiated, I/O actions proceed independently of the AU's until completed.

Two transformer rectifier sets are used to convert the primary ac voltages to secondary dc voltages. A redundant power distribution capability is provided internal to the CP. Each set contains power circuitry in active redundancy to be able to use either of the secondary sources.

It was noted earlier that the state-of-the-art in smaller aerospace computers is well advanced for the type needed for the preprocessors. The typical characteristics achievable from these is shown in Table 1-7.



*critical functions

(a small portion of IO/AU overhead - that which services the critical 9.8 KEAPS of the AU - can be considered critical)

NUMERICAL UNITS ARE KEAPS

Figure 1-9. Detailed CP Subsystem Functional Allocations

Table 1-6. Technical Characteristics of the Central Processor

Type:

Multiprocessor, conventional organization, parallel, binary, 16/32 bit data and instruction words.

Operating Memory, M-2:

Two required, plated wire, NDRO, each consists of five memory modules of 13K x 33 bits maximum, one parity bit per memory word, one parity word exclusive ORed with block address for every five memory words, echo checking of write operations, one microsecond cycle time with interleaving of the five memory modules, maximum capacity of 18K x 33 bits per each module.

Auxilliary Memories:

Mass Memory - M3, Virtual memory using paging methods, error detection using one parity bit per word and one parity word with address exclusive ORed per every four data words; echo checking of write operations, 2 mil plated wire, NDRO, maximum capability of 1280K x 33 bits, modular design based upon 64K modules.

Archive Memory - Magnetic tape storage with $>5 \times 10^6$ bits per cartridge.

Input-Output:

Two required, each contains dual I/O units with comparator AU initiated with self-contained control, solid state buffer memory of nominal 2K x 33 words and 200 nanosecond cycle time, interface with Data Bus Control Unit, Telemetry Bus, and Mass Memory.

Arithmetic Set:

Two required, each contains dual arithmetic units with comparator, solid state buffer memory of nominal 2K x 33 words and 200 nanosecond cycle time 1 million equivalent adds per second per set, fixed and floating point with 100-200 instructions.

Physical Estimates:

	Mass Memory	Archive Memory	Multiprocessor Set
Size, cubic inches	3900	1200	1000
Weight, pounds	180	40	290
Power, watts	15	45	400

Table 1-7. Technical Characteristics of the Preprocessor

Type:

Uniprocessor, parallel-binary, 16 bits data, 16/32 bit instruction words

Memory:

Capacity - 20K word, 17 bit Plated Wire Storage.
One bit of parity per 16 bits.
Cycle time - 1 microsecond

Input/Output:

One buffered 16 bit parallel input and output channel.
Eight external interrupts.

Instruction Repertoire:

Single and double word addressing.
Single word non-addressing.
Indexing
Indirect addressing.

Add Times (Fixed Point):

Add - 4 microseconds
Multiply - 20 microseconds
Divide - 40 microseconds

Special Features:

Internal and external interrupts
General register file usable as index, base or data register

Physical (20K x 17 Bits):

Size - 400 cubic inches
Weight - 15 pounds
Power - 50 watts

The physical values shown in Tables 1-6 and 1-7 are achievable with today's packaging capability. The processors are based upon the use of cased devices on multilayer boards. The mass memory utilizes 2 mil plated wire and power strobing and high density devices with beam leads, hybrid thin film, and ceramic substrates.

1.3 DPA ENGINEERING MODEL DEVELOPMENT PLAN

The objective of Engineering Model Development Plan task was to provide the specification of the Engineering Evaluation Model (EEM) processor and its development plan for the central control of the Data Management System (DMS). The DMS consists of hardware and software which is being assembled and tested for operation in the time period of 1973-1984 by NASA, MSC-Houston, to study data management problems associated with the Space Station and Shuttle programs. The DMS is being configured such that its operation approximates the operation of the Modular Space Station Data Processing Assembly. The DMS currently is defined as consisting of:

- a. DMS Processor
- b. Data Bus Control Unit (DBCUs)
- c. Digital Data Bus (DDB)
- d. Remote Acquisition and Control Units (RACUs)
- e. Preprocessors

The development plan identifies the tasks for the analyses, fabrication, and evaluation of a breadboard processor configuration. Technology is not specified. The end product of the development effort will be a processor that has been integrated and tested to function with the existing elements of the DMS.



2.0 REQUIREMENTS ANALYSIS

2.1 BASELINE DATA PROCESSING ASSEMBLY REQUIREMENTS

2.1.1 Technical Objective

The objective of this IMS Advanced Development task was to develop a preliminary set of parametric data which defines the modular space station (MSS) subsystems support and interfaces which must be provided by the data processing assembly.

2.1.2 Background

The requirements for long duration manned spacecraft include continuous maintenance of operational capability with minimum crew participation. This requirement can be achieved by automating operations of subsystem functions and managing their performance to achieve an integrated base for scientific operations. For the space station and subsequent spacecraft, the automation will require implementing a computer system. The computation required may be performed either by a processor dedicated to a function (similar group of functions), or a large capacity general-purpose computer. The concept, performance, mechanization, reliability, and cost of this capability is sensitive to the specific requirement as defined by the automation required and the interaction between a computer assembly and the operational subsystems.

2.1.3 Scope

The scope of this study was to define in preliminary form the DPA data input/output flow rates and traffic flow patterns, allocation of logical and computational functions which would provide the basis for the development of information flow diagrams, and initial definition of a DPA configuration for data thruput authority simulation.

2.1.4 Study Approach

Three other studies have been performed which have provided input data to this study: NR IR&D study, "Automatic Control and On-Board Checkout (SD 71-227), Autonetics Guidance and Control Subsystem Study (NASA Contract NAS9-10416), and IBM On-Board Checkout Study (NASA Contract NAS9-11189). The approach of this study was to (1) define the subsystems' functions which require data processing support, (2) define the mechanization required to provide data processing support for each function identified in (1) above, (3) estimate the memory, speed and input/output data rates required for mechanization of each function, and (4) integrate the subsystem computation requirements to define a total set of MSS DPA requirements.

Figure 2-1 is a logic diagram showing the NR study approach. The dotted blocks show related studies and other reports which contain data pertinent to the study.

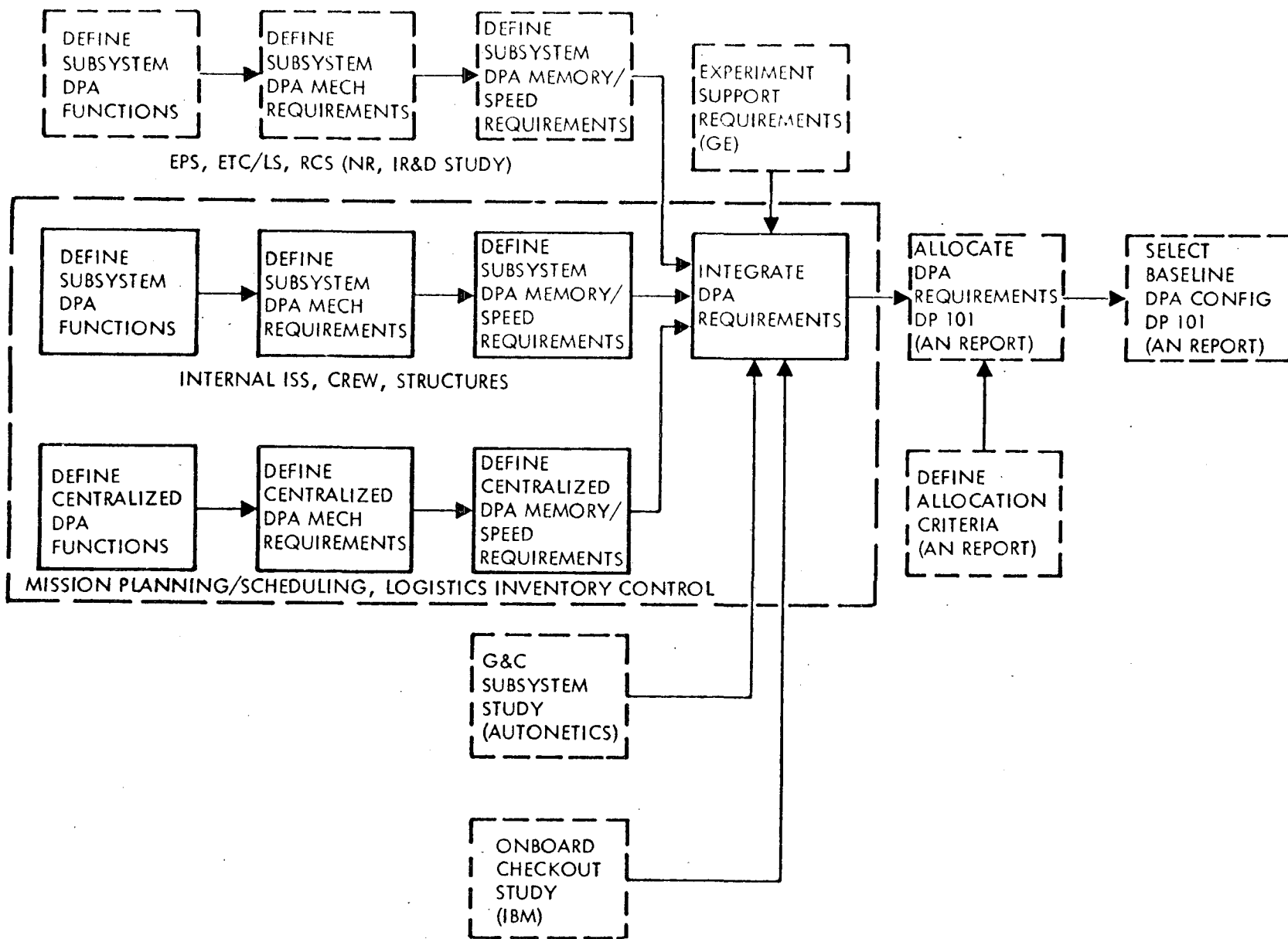


Figure 2-1. Requirements Analysis Study Approach



2.1.5 Guidelines

The following guidelines were used as the initial frame of reference for this study.

1. The Modular Space Station System Requirements Book (SD 71-205) was used as the basis for the space station requirements and build-up configuration.
2. The initial DPA configuration was as specified in DRL-60, Shuttle-Launched Modular Space Station (SD 70-546).
3. The station operations central processor shall provide backup capability for the experiments central processors and vice-versa.
4. Sizing of the station operations central processor shall be determined solely by the station operations computational support requirements (directed by NASA).
5. The central processor has multiprocessing and multiprogramming capabilities.
6. Subsystems computations and logical operations will be performed at the lowest level processor (i.e., preprocessor--lowest level, central processor--highest level) to the maximum extent possible.

2.1.6 Summary of Results

This section summarizes the parametric results of the Subsystem Input/Output Interface Study; no attempt will be made to establish the study rationale here, but rather to report the highlights of the study.

As stated previously, the primary objective of this study was to determine the computation support that the data processing assembly (DPA) must provide to the other MSS subsystems in order that the subsystem functions can be performed. For that reason, the parametric interface data that were developed consisted of the computational requirements that the DPA must provide to each subsystem and the amount of data flow that must occur across the interface between the DPA and each subsystem. While the nature and number of interface signals were defined in order to size the interface data rate, no attempt was made to establish the physical aspects of the interface such as number of wires, signal levels, formats, etc.

The purpose for making this study was to generate the data necessary to size the data processing assembly and permit the definition of a DPA configuration. The DPA in this study includes the central processors, preprocessors, RACU's and digital data bus.



There were four primary interface parameters that were developed: (1) the software or program size required in the DPA to support a subsystem; (2) the speed at which each program must be executed; (3) the rate at which sampled data must be input from a subsystem to the DPA; and (4) the rate at which computed data must be output from the DPA to a subsystem. The program size parameter translates directly into DPA memory size requirements and the unit of measure is 32 bit words. The speed parameter is a measure of the number of instructions that must be executed per unit time and the dimension is equivalent adds/second. Both the input and the output rates are measures of the data flow across the interface dimensioned in digital bits/second. Note that the terms "input" and "output" as used in this study are referenced to the DPA, i.e., data are "input" to the DPA from a subsystem and "output" from the DPA to a subsystem.

For ease in estimating the subsystem support requirements that the DPA must provide, the tasks were broken down into computation functions that the DPA must perform. In many cases, these computation functions were the same as the subsystem functions that were defined in the Phase A study. For example, in the G&C subsystem functions such as attitude determination, navigation determination and control moment gyro (CMG) control were used. In other cases, it was necessary to define new computation functions such as deploy solar array booms, solar array pointing control, and fuel cell control for the electrical power subsystem. The DPA to subsystem interface requirements were developed for each of these computation functions in terms of the parameters discussed above and then treated in a linear fashion to reach the total interface requirements.

A summary of the DPA interface requirements is presented in Table 2-1. The first column lists the subsystems that require computational support and the other functions that are required by the DPA in order to provide that support. The first section of the table lists the programs or memory sizes and shows where they predominantly reside, that is, in operating (rapid access time) memory or mass (medium access time) memory. Archival memory (magnetic tape) is required as backup for all of the programs that will normally reside in operating and mass memories in order to provide a "refill" capability in case of a malfunction and/or when replacement is made. In addition, a large amount of archival storage is required for the data base and for storage of experiment data. The total memory requirement is estimated at 495 thousand words (32 bits) of operating memory, 1.3 million words of mass memory and 13 million words of archival memory.

The second section of Table 2-1 tabulates the speed at which each program must be executed. The EPS is the major contributor to the overall speed requirement with 7 million equivalent adds/second. The total speed requirement is 8 million equivalent adds/second; however, this figure is not too meaningful because it implies that all programs will be executed concurrently. The only real merit of this total speed requirement is to form an upper boundary or worst case speed condition.

The final section of Table 2-1 deals with the DPA input and output interfaces. The number and types of signals are listed along with the rate at which data will flow across the interfaces. The total input data rate is 4 million

Table 2-1. DPA Interface Requirements

Subsystem	Memory (32 Bit Word)			Speed (EQ Adds/Sec) (K)	DPA Input Interface				DPA Output Interface		
	Operating (K)	Mass (K)	Archive (K)		Analog Signals	Discrete Signals	Digital Signals	Rate (Bits/Sec) (K)	Dig. Sig.	Dis. Sig.	Rate (Bits/Sec) (K)
G&C	13.3	10.2	23.5	225.3	-	96	205	57.3	209	96	71.6
EPS	71.9	0.8	72.7	7000.0	5990	1348	-	1868.3	69	4.5	1529.0
ETCLSS	2.3	4.5	6.8	28.5	505	320	-	11.7	24	312	890.0
RCS	0.9	-	0.9	10.9	75	69	-	5.2	2	46	432.0
Crew		8.2	8.2	1.0	-	3	22	1.3	14	0	1.1
Structure	-	305.0	305.0	786.0	-	708	1	1.1	12	80	192.0
ISS Communications	8.0	-	8.0	30.0	18	166	236	5.9	134	32	2000.0*
ISS Controls and Displays	2.2	14.5	16.7	13.7	-	200	600	1.6	600	0	12.8
Mission Management	23.0	62.1	85.1	30.2	-	-	-	82.3			96.0
OBCO	27.3	145.7	237.0	147.8**	7408	2853	250	** 109.0	-	-	3.8**
Data Base	-	582.5	3950.0	-	-	-	-	-	-	-	-
DPA Executive	22.2	36.9	59.1	-	-	-	-	-	-	-	-
Experiment Support	324.0	397.0	8900.0	615.0	-	-	-	2000.0	-	-	*
Totals	495.1	1262.7	13,368.3	8103.2				4143.7			3724.8
*Communication downlink \approx 2 million bits/sec from experiments											
**Single worst case (EPS)											

2-5

SD 72-SA-0114-4

bits/second and the total output data rate is 3.7 million bits/second, with the EPS and experiments being the major contributors.

As with the speed requirement, these total data rates represent what would be required if all tasks were performed concurrently and should only serve as upper bounds on the interface data rates. No allowance was made for identification or formatting of data for internal DPA data flow. The thing that is of primary interest is the rate for which the DPA digital data bus must be designed in order to carry the input/output data without reaching saturation. To answer this question, both the interleaving of programs and configuration of the DPA itself had to be considered. Although this was beyond the scope of this study, it was felt that a preliminary analysis should be conducted to establish a baseline bus data rate. The analysis was conducted and the resultant preliminary data bus operating rate requirement is 6 megabits/second.

2.1.7 Conclusions

This study has shown that the DPA, which includes the central processors, preprocessors, RACU's and digital data bus, must possess the following characteristics in order to support six-man station operations and experiment operations.

Operating memory	495K words
Mass memory	1.3M words
Archive memory	13 words
Speed	8M equivalent adds/second (worst case)
Data bus I/O rate	6 megabits/second

These characteristics must be considered within the constraints and limitations that (1) redundancy to meet the failure criteria has not been included, and (2) experiment requirements are still very preliminary. The full impact of these two factors on the DPA characteristics is not intuitively obvious at this time.

The two major contributors to the DPA computation and data rate requirements are the electrical power subsystem (EPS) and experiments. The EPS requirement is caused by the highly automated design approach and the need to react to an electrical power overload or fault condition within 4 milliseconds. Experiment support requirements are largely caused by the high rates associated with image sensors.

To say that the DPA requirements are large is not really enough. These requirements need only be compared to the Apollo computer (39K memory and 43K adds/second speed) to gain a perspective of the MSS DPA complex. A more realistic comparison is the IBM 360-85 located at Space Division's Downey facility which is used on a time-share basis to perform data processing for all of NR's Southern California Divisions. The 360-85 has an operating speed of 650K equivalent adds/second, an operating memory of one million bytes and a mass memory of four million bytes. In comparison the DPA requires the size and a mass memory slightly larger.

It is german here to mention the baseline DPA configuration that is presented in Section 3.0 of this volume. The parametric data generated in this study was largely used to define that baseline configuration. That configuration currently consists of 26 computers to handle the computation load: 24 preprocessors that have been allocated high-speed dedicated computations; one multiprocessor for generalized station operations computations and overall supervision of the other processors; and another multiprocessor for experiment operations. So, while the DPA computation requirements summarized above are large, they can still be satisfied with computers that exist today and new technology developments are not required. However, many advanced architectural concepts are required, and these have been studied as indicated in the following sections of this volume.

2.2 REFINEMENTS TO BASELINE REQUIREMENTS

A baseline DPA configuration was selected as a reference point for successive tasks (see Section 3.0). That configuration consisted of a central processor (CP) and 24 remote processors (RPU). The RPU's were allotted the high-speed computations; the low speed and supervisory computations were assigned to the CP (this also reduces the data bus traffic).

2.2.1 Modifications to Basic Requirements

Figure 1-3 shows the ADT tasks and, partially at least, the impact of the simultaneous MSS Phase B Study. The Phase B requirements, as they were understood at the beginning of the study, were the major input to the ADT study. Based on these requirements, a baseline DPA configuration was selected. This baseline configuration was used in both the ADT tasks and the Phase B studies. Naturally, the basic requirements reflect the preliminary nature of our understanding of the MSS subsystems. The continuing analyses of the MSS subsystems during the Phase B studies had the effect of reducing the computational requirements for subsystem support.

Figure 2-2 picks out the sequential nature of the ongoing requirements analysis. Almost all of the difference between the two requirements summaries results from a reduction in the computational support required by the electrical power subsystem (lowering of the 4 msec response requirement for 90 percent of the functions) and from the placing of the telemetry/command data on a separate dedicated bus. Lastly, no experiments requirements are reflected in the Phase B requirements summary.

2.2.2 Design and Growth Margin Philosophy

The DPA computation requirements have been expressed in terms of five parameters: (1) processing speed, (2) operating memory size, (3) mass memory size, (4) digital data bus rate, and (5) archive memory size, as shown in Table 2-2. The basic requirements for these computation parameters were determined by trial programming the various functions allocated to each MSS subsystem.

INITIAL MSS REQUIREMENTS

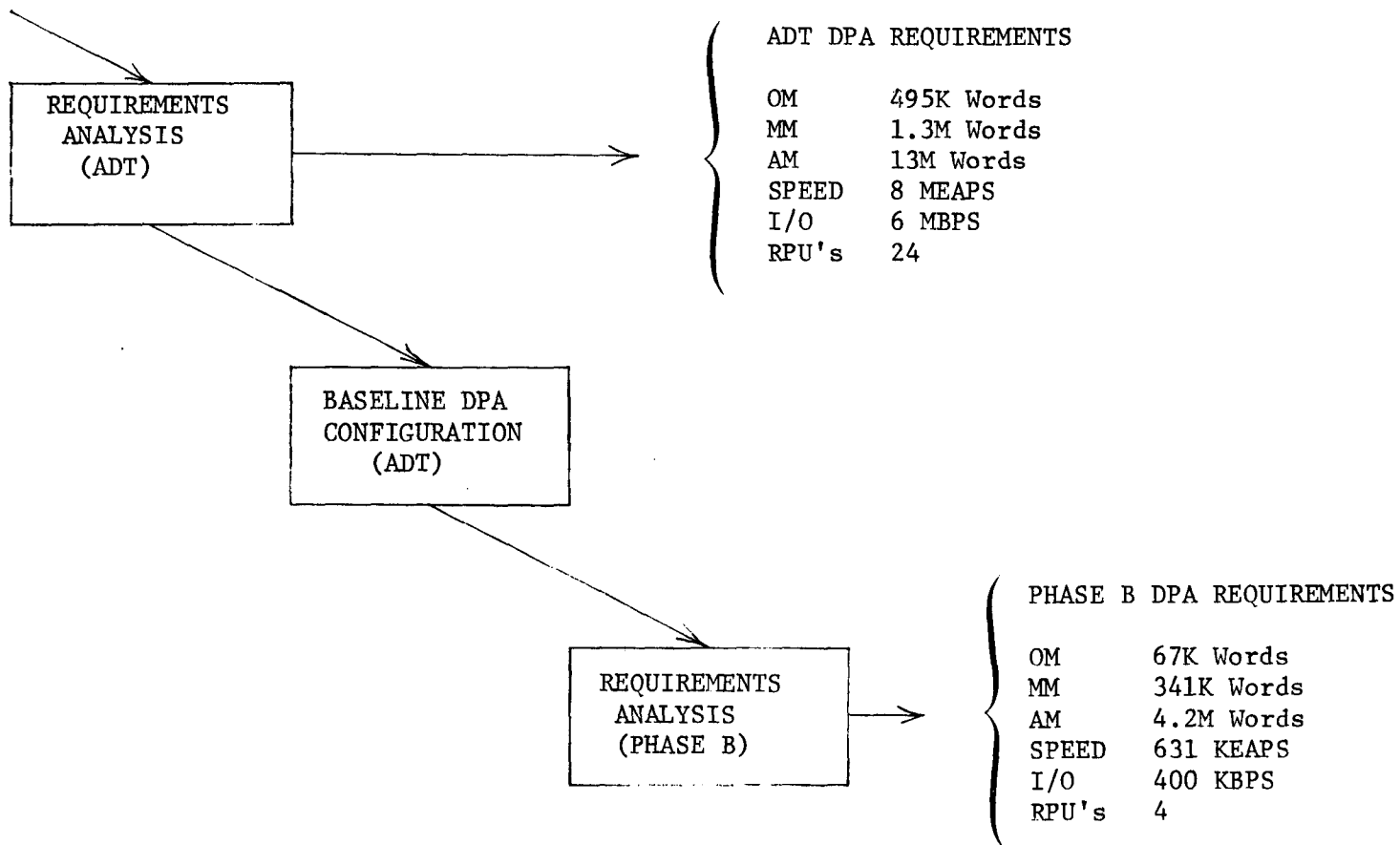


Figure 2-2. Sequential Analyses of DPA Requirements

Table 2-2. Computation Requirements for Station Operations

	Performance Requirements	Base Requirements		Maximum Requirements		Initial (6-Man) Requirements	
		Basic Requirements	Design Margin (100%)	Maximum Growth Margin (100%)	Maximum Design Requirements	Initial Growth Margin	Initial Design Requirements
Central Processor	Processing Speed (equivalent adds/second)	631K	631K	631K	1893K	0	1262K
	Operating Memory (32 bit words)	67K	67K	67K	201K	0	134K
	Mass Memory (32 bit words)	341K	341K	341K	1023K	0	682K
	Data Bus Rate (bits/second)	400K (Station Operation)	400K (Station Operation)	7.2M*	10M	7.2M	10M
		2000K (experiments)					
	Archive Memory (32 bit words)	4.2M	4.2M	4.2M	12.6M	0	8.4M
Remote Processing Unit	Processing Speed (equivalent adds/second)	125K	125K	0**	250K	0	250K
	Memory (32 bit words)	9K	9K	0**	18K	0	18K
<p>*Special allowance for experiments</p> <p>**RPU growth will be accommodated by additional units</p>							

Design margin is that factor that was added to the basic computation requirements because of uncertainty in estimating the basic requirements. Uncertainty arises from many sources such as lack of definition of the functions that a subsystem must perform, estimates for the control algorithms, and assumptions pertaining to processor characteristics. A design margin of 100 percent of the basic requirements was used in the Phase B study.

Growth margin is that factor that was added to the basic computation requirements for (1) planned growth from the six-man station to the 12-man station, and (2) uncertainty of future mission requirements. A DPA "modularity" design philosophy is intended to permit computation capability add-on (expandability), and for that reason the growth margin was minimized for the initial six-man implementation. The digital data bus rate is the only DPA parameter that cannot be designed with a growth capability and for that reason a growth margin of 7.2 megabits/second has been included at the outset. This is an estimate of maximum data rate capability (10 megabits per second) that will ever be required. The other computation parameters do not have a growth factor included in them because of the expandability feature of the DPA design.

2.2.3 Final DPA Requirements

As shown in Table 2-2, the design requirements are a summation of the basic requirements, design margin, and growth margin. The design requirements represent the size and performance characteristics that were used in implementing the DPA design.

In arriving at the DPA computation requirements and resultant implementation, the following general philosophy or set of ground rules was used:

1. The design margin shall be 100 percent of the basic requirements.
2. A growth capability shall be provided; however, a growth margin shall not be included in the initial (six-man) requirements, except in those cases where growth by expandability cannot be accommodated.
3. A digital data bus rate of 10 megabits/second is the maximum data transfer requirement for both station operation and experiment operation.
4. The experiment operation central processor and memories will be duplicates of the station operation hardware.

3.0 BASELINE DPA CONFIGURATION

There are numerous alternatives which might be considered in the preliminary DPA characteristics study task. In enumerating these alternatives, a systematic approach has been taken to assure that all alternatives receive consideration. In view of the modular nature of the MSS, there are three distinguishable data processing applications: overall station data processing, module (local) data processing, and subsystem/function dedicated data processing.

The following assumptions have been made to begin the enumeration of alternatives:

1. Some kind of overall station data processing is required; this may be either a uniprocessor (UP), a multiprocessor (MP), or a multicomputer (MC).
2. At the subsystem/function dedicated data processing level, consideration will only be given to the presence (P) or absence (O) of data processing; the particular type and quantity of processing required by a function or a subsystem will be individually determined.
3. At the module data processing level, there may be either an MC, an MP, a UP, or no processor at all (O).

Based on these assumptions, there are 24 alternatives, as shown in Table 3-1.

3.1 DISTRIBUTION OF SUBSYSTEMS REQUIREMENTS

Since the space station is modular, the distribution of processing requirements by module is needed in selecting a DPA configuration. An assumption was made, based on the subsystems physical distribution, of the distribution of signals associated with the required computing functions. Table 3-2 lists these functions by subsystem and indicates the assumed distribution within the station modules.

3.2 CONFIGURATION TRADEOFFS

On the basis of the assumptions established for this preliminary DPA configuration study for the modular space station, a number of basic features can be established.

Pre-processors are required since frequent or cyclic computing tasks exists. The pre-processors are to be determined by subsystem functional needs and can be either multiprocessor, multicomputing or uniprocessors.

Table 3-1. DPA Alternatives for Modular Space Station

DPA Alternative	Overall Station Data Processing	Module Data Processing	Subsystem/ Function Data Processing
1	UP	O	O
2	UP	O	P
3	UP	UP	O
4	UP	UP	P
5	UP	MP	O
6	UP	MP	P
7	UP	MC	O
8	UP	MC	P
9	MP	O	O
10	MP	O	P
11	MP	UP	O
12	MP	UP	P
13	MP	MP	O
14	MP	MP	P
15	MP	MC	O
16	MP	MC	P
17	MC	O	O
18	MC	O	P
19	MC	UP	O
20	MC	UP	P
21	MC	MP	O
22	MC	MP	P
23	MC	MC	O
24	MC	MC	P

Table 3-2. MSS Signal Allocation Assumptions

Subsystem	Function	Percent Per Module							
		Power	Core	SM1	Cargo 1	SM2	SM3	SM4	Cargo 2
ETC/LSS	Pumpdown and Repress		100						
	CO ₂ Management					50	50		
	Electrolysis Control					50	50		
	O ₂ Partial Pressure					50	50		
	Humidity & Contamination Control					50	50		
	Circ. & Temp. Control	16-2/3	16-2/3	16-2/3		16-2/3	16-2/3	16-2/3	
	O ₂ /N ₂ Control					50	50		
	Active Thermal Control	50		12.5		12.5	12.5	12.5	
	Humidity & Urine Rec. Cont.					50	50		
	Wash Water Recovery					50	50		
	Food Management					50	50		
	Special Life Supplies		16-2/3	33-1/3		16-2/3	16-2/3	16-2/3	
EPS	Deploy SA Booms	100							
	Retract SA Booms	100							

Table 3-2. MSS Signal Allocation Assumptions (continued)

Subsystem	Function	Percent Per Module							
		Power	Core	SM1	Cargo 1	SM2	SM3	SM4	Cargo 2
EPS (continued)	Extend SA Panels	100							
	Solar Array Pointing Control	100							
	SA Inverter Control	100							
	Battery & Fuel Inverter Control	50	50						
	Battery Charging	50	50						
	Primary Power Bus Control	50	50						
	Secondary Power Bus Control	50	50						
	Fuel Cell Control		100						
	SSCB Control	10	5	30	2.5	10	10	30	2.5
	Differential Current Meas.	45	45	1-2/3	1-2/3	1-2/3	1-2/3	1-2/3	1-2/3
	Lighting Control	10	10	15	10	15	15	15	10
RCS	Nitrogen Quantity Balance		100						
	Hydrogen Gas Control	100							
	Hydrogen Cryogenics				100				

Table 3-2. MSS Signal Allocation Assumptions (continued)

Subsystem	Function	Percent Per Module							
		Power	Core	SM1	Cargo 1	SM2	SM3	SM4	Cargo 2
RCS (continued)	Thrust Valve Control	50	50						
	Oxygen Gas Control	100							
	Oxygen Cyrogenics				100				
Structures	Berthing	16-2/3	83-1/3						
G&C	IRU Functions	100							
	ORU		100						
	CMG Control		100						
	RCS Electronics	100							
	Exp. Module Update							100	
	Shuttle Alignment			100					
	Terminal Rendezvous			100					
	Berthing Control			100					

3-5

SD 72-SA-0114-4

Table 3-2. MSS Signal Allocation Assumptions (continued)

Subsystem	Function	percent Per Module							
		Power	Core	SM1	Cargo 1	SM2	SM3	SM4	Cargo 2
Crew	Real Time Medical Data Acquisition						100		
	Non Real Time Medical Data Acquisition						100		
	Medical Data Analysis						100		
ISS	Internal Communications Control			100					
	External Communications Control	30	20	50					
	Tracking Control							100	
	CMD and Message Generation			100					
	Displays and Controls			100					
	Subsystems Operations			100					
	Planning and Scheduling			100					
	Logistics Inventory Control			100					
	Information Storage and Retrieval			100					

Table 3-2. MSS Signal Allocation Assumptions (continued)

Subsystem	Function	Percent Per Module							
		Power	Core	SM1	Cargo 1	SM2	SM3	SM4	Cargo 2
ISS (continued)	Mission Analysis and Assessment			100					
	Record Management			100					
	Printer Control			100					
	Remote Terminal	33-1/3		33-1/3				33-1/3	
	OBCO G&C	Op=100 Mass 50	Mass 50						
	OBCO RCS	40	40		20				
	OBCO EPS								
	SSCB	10	5	30	2.5				
	Other	47	47	1	1	1	1	1	1
	OBCO ETC/LSS	15	5	10		30	30	10	
	OBCO Ext. Comm.	10	25	50				15	
	OBCO Int. Comm.	25	10	50			15		

3-7

SD 72-SA-0114-4

Table 3-2. MSS Signal Allocation Assumptions (continued)

Subsystem	Function	Percent Per Module							
		Power	Core	SM1	Cargo 1	SM2	SM3	SM4	Cargo 2
ISS (continued)	OBCO DPA	25	25	25	2.5	5	5	10	2.5
	OBCO Compiler			100					
	OBCO Tables	25	25	2.5		5	5	10	2.5
	OBCO Structures		100						

With regards to the candidates listed in the previous Section 3.0, the selection of a multiprocessor at the central level can be made for the following reasons:

The Station A results obtained in a GE study for the selection of a multiprocessor for a central computer was reviewed and felt to be valid. While the computational load on the central computer is reduced for the Modular Space Station over the Station A, the computer requirements as noted from the central computing functions of OBCO and mission/planning still are large. Hence the conclusion is that a multiprocessor is better for reasons of size and weight, power, expandability, system flexibility, requires excessive components for equivalent performance or backup and introduces complexity in control. The uniprocessor would require duplication of a large amount of memory for the redundancy consideration and be less flexible or expandable with regards to future requirements. Table 3-3 presents an evaluation of the central computer parameters. While no weighting is given to these, it is evident that multiprocessing would be the best selection. (Note: The class of computer organization which contains multiple, modular elements while still having a single central processing unit in operation but providing reconfigurability is classified here under multiprocessing.)

At the module processor level, the multicomputer can be eliminated since, if the computing load is large, this organization has no advantage over the multiprocessor by the above reasons. Further, since the selection of the multiprocessor at the central level has been made, for commonality reasons, which relate to cost effectiveness, the multiprocessor would be a better selection. If the computing load is small, the uniprocessor is better for commonality with the preprocessors. This selection would require duplication of the uniprocessor in the first modules in order to meet the reliability dictate of Fail Operation, Fail Safe during buildup. Final selection of the uniprocessor or multiprocessor for the module level is dependent upon the local computational load being large or small.

The candidates for further analyses then become:

<u>DPA Alternative</u>	<u>Central</u>	<u>Module</u>	<u>Subsystem</u>
A	MP	O	P
B	MP	MP	P
C	MP	UP	P

At this stage, the decision whether to have module computing or not was answered by the consideration of the functions performed by those processors. These are module supervisory control and checkout during ground test and station buildup. The ground test can be done with a ground based computer connecting into the data bus. This connection is probably required anyway in order to check the data bus. Control during buildup can be done without module computers by locating the central processor in the first-launched module and using it to control the buildup operations. An alternate to this is to provide a command/telemetry communications-link to the ground and use ground equipment to establish supervisory control and checkout until the central computer arrives in SM1.

Table 3-3. Central Processor Selection

Criteria	Uniprocessor	Multicomputer	Multiprocessor
Hardware Complexity	1	2	3
Software Complexity	2	2	1
Size, Weight	2	1	3
Power	1	1	3
Expansion Capability	1	1	3
System Flexibility	1	1	3
Graceful Degradation	1	1	3
Development Status	1	2	2
Logistics	3	1	3
On-Board Checkout	2	2	2
Test and Validation	2	3	2
System Reliability	2	1	3
Cost	1	2	3
Rating: 0 = No Impact 1 = Poor 2 = Medium 3 = Good			

Since there were reasonable ways to provide the computing functions at the module level, the decision for the preliminary DPA was that alternative A be selected. Further, of the two variations, i.e., placing the central computer in the first-launched module or providing a communication's link control during buildup, the latter was selected based upon system considerations which include the reduction of weight and other physical attributes in the first modules and the desire to keep the central computer in SMI. This desire stems from the interface considerations with the central computer peripherals (mass memory, displays and control console, printers, etc.) and the consideration of bringing the SMI module back for modification or repair. In this event, an integral command and control module has advantages.

3.3 BASELINE DPA CHARACTERISTICS

The baseline configuration is the multiprocessor central processor located in the Primary Control Module (SM1) with pre-processors distributed as required plus buildup supervisory control provided by a Buildup Command Control Data Processor (BUCCDP) located in the first-launched module to provide commands for both the power and core modules during buildup only. This component will be removed or disengaged when SM-1 arrives and supervisory control transferred to the stations operation central processor.

Several methods exist for linking the BUCCDP with the DPA. The baseline concept is to provide a separate redundant bus to each of the pre-processors and RACU's in the power and core modules. An alternate way would be to connect the BUCCDP to the Data Acquisition and Control Subassembly. The first method requires additional busing and affects the RACU and pre-processor interface design. The latter approach requires a combination of Data Bus Control Unit, communication command demodulator/telemetry link, and some means of controlling the combination. Since the object is to minimize hardware in the first modules, it was concluded that a redundant hardware bus network is a more effective way. Further study in this area is needed.

Figure 3-1 gives a block diagram of the baseline ADT DPA showing the number and distribution of pre-processors and RACU's. Further shown is the interconnection of the peripherals within the SM1 and SM4 modules.

Safety of operation is provided through use of redundancy of equipment and location. The two control centers are located in two separate pressure volumes. Interconnection is provided with a multiple bus network. Maintenance is facilitated with an OBCO system which includes the monitoring of signals and the ability to isolate faults with either automatic or man-generated checkout programs. Other features of this preliminary DPA are commonality arising from similar components and few types; flexibility due to the bus structure; incremental buildup capability and interchangeability of components; and operational availability due to several levels of redundancy and degraded modes of operations.

Components of the DPA are realizable with the present aerospace state-of-the art. Future improvements in physical characteristics and cost are possible with the expected technology changes in memories (solid-state and plated wire) and logical devices.

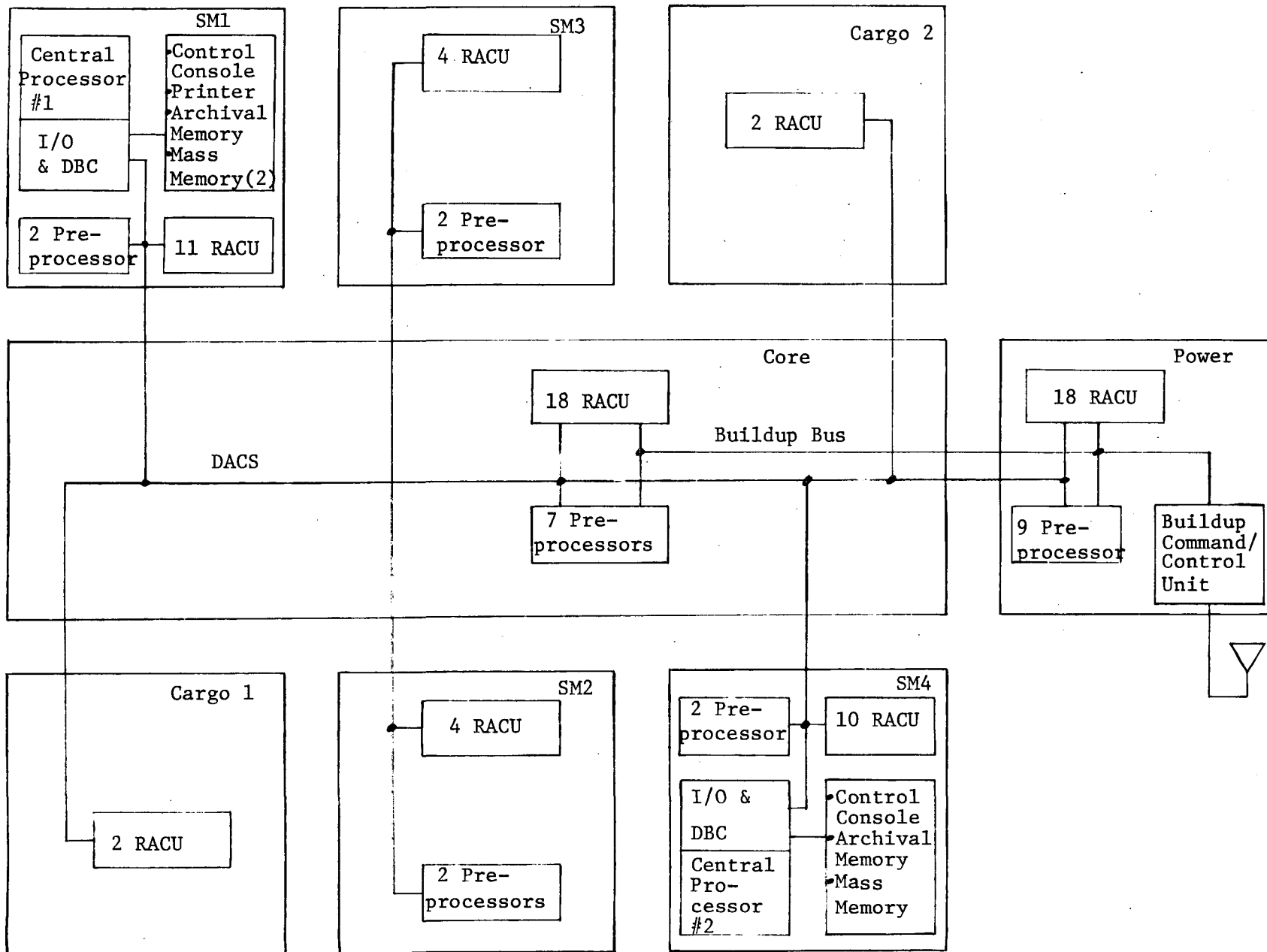


Figure 3-1. Baseline DPA Block Diagram

The baseline configuration also offers the system advantage of permitting subassemblies and subsystems to be operated and checked out prior to total system integration. The approach of using pre-processors should be cost-effective in eliminating special purpose circuitry at lower system levels required otherwise.

The central computing complex supervises the pre-processors and controls the communication with the space station and ground subsystems. It supplies the spacecraft and mission management operation and overall fault isolation plus crew interface. The central processor has access to the measurements and control points within the affected subsystems through the RACU's. Redundancy of monitoring of critical signals is done by using the central processor and the pre-processors. Non-critical signals are monitored by the pre-processors only. The central processor controls the overall fault isolation.

The configuration presented here is for a 6-man level. The growth to the 12-man station is accommodated by increasing the memory sizing and adding RACU's to accommodate the increased power load.

The experiment or backup central processor is made identical to the operational or primary central processor. Its normal operation would be to hold critical programs in its operating memory. Periodically data would be supplied to these programs to provide a reference point in the event of reconfiguration for a primary processor failure. The remainder of the computer is devoted to servicing the experiments. Upon reconfiguration, the required operational programs (loaded from mass memory) are performed in addition to the normal experiment support.

Table 3-4 defines the features required for the processors. Pre-processor memory requirements range from 1.8K to 13.2K words. As can be seen from the data in Table 3-4, the pre-processors speed requirements exceed that achievable in state-of-the-art uniprocessors for several of the pre-processors. The functions required involve the repetitive comparison between limits of numerous signals. Autonetics has developed and produced an advanced special-purpose MOS device for this comparison function. The device can result in a reduction of the speed requirements by an amount proportional to quantity used. (A preliminary estimate is that one such device can reduce the speed by a factor of 50.) Further study of the requirements and implementation is suggested.

With special processing, the speed required can be within the range of existing aerospace computers. The sizing for the central processors is included in Table 3-4.

NOTE: This baseline configuration served two uses: as a configuration for further study in the ADT effort; and as a configuration for further study in the MSS Phase B study. As a result of this second use, a design decision was made as part of the Phase B study to lower the processing requirements (see paragraph 2.2.1) and to standardize the design of the



Table 3-4. Summary of Computer Sizing

Ground Rules: Standard List - OBCO provided during buildup. Battery charging allocated to preprocessing. Memory estimation given in 32 bit words.					
Module	Computer	Operating Memory	Mass Memory	Speed Ops/Sec	Archival Memory
Power ⁽¹⁾	Preprocessor 1	4.0K		1104K	
	Preprocessor 2	4.0K		1104K	
	Preprocessor 3	1.8K		295K	
	Preprocessor 4	1.8K		295K	
	Preprocessor 5	13.2K		65K	
	Preprocessor 6	7.5K		741K	
	Preprocessor 7	7.5K		741K	
	Preprocessor 8	9.1K		125K	
	Preprocessor 9	9.1K		125K	
Core ⁽¹⁾	Preprocessor 10	2.5K		807K	
	Preprocessor 11	2.5K		807K	
	Preprocessor 12	13.2K		65K	
	Preprocessor 13	6.7K		665K	
	Preprocessor 14	6.7K		665K	
	Preprocessor 15	2.6K		40K	
	Preprocessor 16	2.5K		1K	
SM1	Preprocessor 17	5.0K		475K	
	Preprocessor 18	5.0K		475K	
SM2	Preprocessor 19	1.9K		173K	
	Preprocessor 20	1.9K		173K	
SM3	Preprocessor 21	1.9K		173K	
	Preprocessor 22	1.9K		173K	
SM4	Preprocessor 23	5.0K		475K	
	Preprocessor 24	5.0K		475K	
SM1/SM4	Central-Subtotal	74.0K	259.6K	581K	
	-Transient Memory	16.0K			
	-Support Package		98.0K		36K
	-Data Base		248.4K		2360K
	-OBCO				64K
	-Master Backup				804K
		90.0K	606.0K	581K	3264K

(1) Requires Ground Backup Command and Control

units making up the MSS subsystems. Hence, as will be shown in Section 9, the later configuration will use one type preprocessor (sized to the maximum preprocessor requirement) and one type RACU (which is, however, modularly incrementable as required).

4. INFORMATION FLOW STUDY

4.1 INTRODUCTION

The purpose of the information flow study was to define the MSS DPA information flow so that the DPA may be simulated using NASA's IMSIM for DPA thrupt analysis. The simulation of the DPA in turn will be used as a tool to assist in defining the final DPA configuration selection. The basis for the study has been the results of related Space Station studies, and the overall description of the Space Station and its mission.

A method of flow diagram presentation and attendant tabulations was carefully selected to provide a comprehensive data file of software and information characteristics that will prove beneficial in the continuation of the Advanced Development Tasks and related studies. This data file consists of descriptions of each subsystem, baseline configuration data, buildup information, DPA computational loads and allocations, computer sizing information, commodity tabulations, signal interface lists, and DPA parametric data requirements.

The decision as to what information would be included in the data file was predicated on the type of information required for a DPA thrupt and authority analysis and the means used to perform this analysis. Two questions that had to be answered before the analysis could be conducted were:

- (1) What is the DPA to do? and
- (2) How is the DPA to accomplish its tasks?

The following rationale was used to gather information to resolve the first question. In order for the MSS to fulfill its orbital mission, certain basic functions have to be performed (e.g, life support; power supply and distribution; experiment preparation, performing, and processing, etc.). The above functions generate some sort of "commodity" (units of information; e.g, data commands, status or anomalies) to be used by the other functions. To facilitate the transfer of these commodities requires a service function of some sort. The DPA provides this service function. The DPA is not a "generator", but rather performs the dissemination, manipulation and storage tasks in regard to the commodities output by the generators on some sort of demand basis (preplanned or dynamic).

The manner in which these generator functions are implemented acts as a requirement on the implementation of the DPA. It is the nature and characteristics of the commodities produced by the generator and the processing to be performed by the DPA in regard to these conditions that determine the capabilities to be included in the DPA.

In effect this rationale dictates the collection of information with regard to the commodities that will enter and leave the DPA, the manipulations that the DPA shall perform on the commodities and the equipment that the DPA will use or interface with to manipulate and transmit these commodities.

The resolution of the second question requires that the collected information be structured on a system basis. The information so structured will indicate the points of entry and exit of commodities, the path over which the commodities are to travel, the location of the manipulation routines through which the commodities are to pass and the transfer function to be performed by the manipulation routine in response to a commodity, the vehicle to be used (i.e., message formats) by the commodities and the gating (message transfer doctrine) imposed on message transfers. In addition, the performance characteristics and configuration of the hardware entities comprising the DPA as well as the doctrine governing the operations of these entities were considered.

4.1.1 Guidelines and Constraints

The guidelines and constraints under which this report was prepared are listed below.

- A. The information flow diagrams depict data flow and configuration allocation for the MSS preliminary baseline configuration (reference Section 3).
- B. The central processor provides central control, checkout and backup for the pre-processors.
- C. The pre-processors provide dedicated computation for a specific function or subsystem.
- D. Module processors have not been considered in preparation of the information flow diagrams. Required centralized processing has been shown under the Station Operations Central Processor flow diagram.
- E. The purpose of a Remote Acquisition and Control Unit (RACU) is to provide a signal interface between the central processor and a subsystem. RACU features are as defined in DRL-13 (SD 70-159-3).
- F. Data transfer between a RACU or pre-processor and the central processor will be accomplished via a serial data bus in order to reduce long wire runs and large signal interface connections between modules.
- G. Subsystem computation tasks which are performed frequently were prime candidates for allocation to pre-processors.
- H. Pre-processor redundancy has been dictated by the redundancy of the subsystem or functional loop that they control.
- I. Two central processors (CP) will be provided, one located in each pressure volume. During normal operations one CP will be assigned tasks associated with station operations and the other will be assigned experiment management and data processing tasks. When one CP has failed, the other CP shall provide a backup capability to

handle critical functions, as a minimum. Backup for non-critical functions will be provided within the constraints of normal memory and speed limitations.

- J. The experiment computation contribution to the central processor flow has not been included.

4.2 DATA PROCESSING ASSEMBLY (DPA) CONFIGURATION

Figure 3-1 gives a block diagram of the baseline DPA showing the number and distribution of preprocessors and RACU's. For this same baseline configuration, an equipment hook-up concept has been developed. Figures 4-1 through 4-6 show this concept for the eight modules of this configuration. These figures show the data bus, the RACU's¹ and the Data Bus Control Units (DBCUs) which make up the DACS. (The data bus also interfaces with the Remote Processing Units (RPU's) just as if they were RACU's). Table 4-1 provides commentary on some aspects of these interconnection diagrams. Also included in these hook-up diagrams are some (interfacing) elements of the MSS Information Subsystem which are not considered to be elements of the DPA. For example, the Remote Terminal Units (RTU) are remote display/control devices driven by the DPA via the data bus; the Modulation Processor is a signal combiner and subcarrier modulator which is part of the Communications Assembly.

4.3 INFORMATION FLOW DIAGRAMS

This section of the report contains the information flow diagrams which show functionally the routes which must be taken by subsystem commodities to accomplish the DPA tasks. These flow diagrams were developed by using the speed, memory and input/output subroutines defined for each subsystem function, tabulating the data on commodity and software sheets with other pertinent information such as iteration rates and concatenations (linkage to other programs) and then drawing a flow diagram which shows the commodity flow and subroutines required to accomplish a specific DPA task.

Information flow lines through the DPA are annotated with the commodity reference numbers which travel over each line. Central processor functions which relate directly with each subsystem is illustrated in the subsystem diagrams shown in Figures 4-7, 4-8, 4-9, 4-10, 4-11, 4-12 and 4-13. All other processor functions are shown on the central processor diagram (Figure 4-14).

Central processor and pre-processor software are identified on the diagrams with numbers in the "S" series. Subsystem equipment and Remote Acquisition Control Units (RACU's) are identified on the diagrams with numbers in the "E" series (for equipment). Information (i.e., "commodities" such as data, commands, status, etc.) which flows between processor functions, RACU's, equipment, and

¹The function and the designation (E-number) of the RACU's shown on the Information Flow Diagrams are not in direct correspondence with the function and designation of the RACU's shown in diagrams 4-1 to 4-6. The former were created first to show a generalized configuration. The latter represent the baseline DPA configuration.

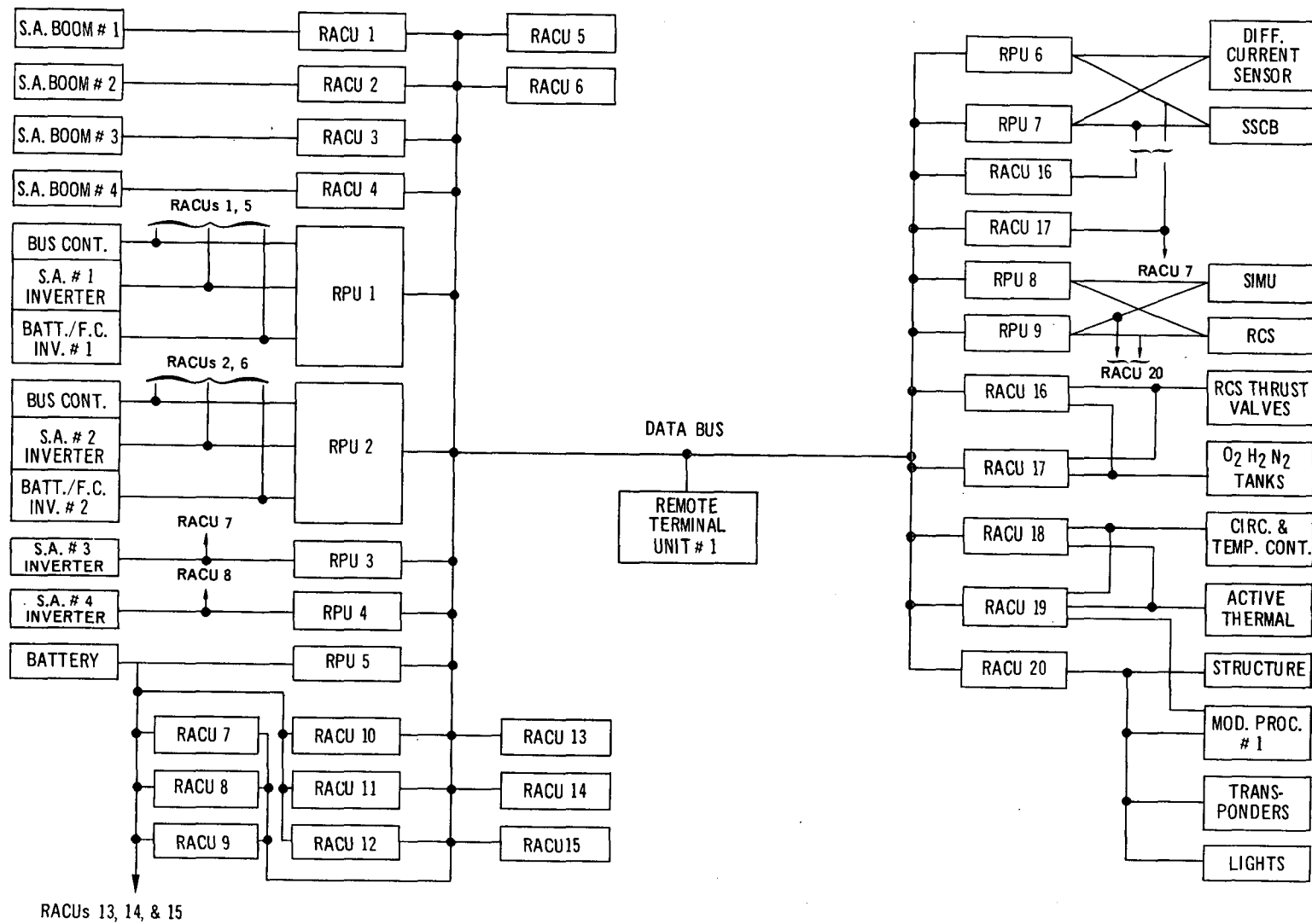


Figure 4-1. Power Module Hookup-Baseline

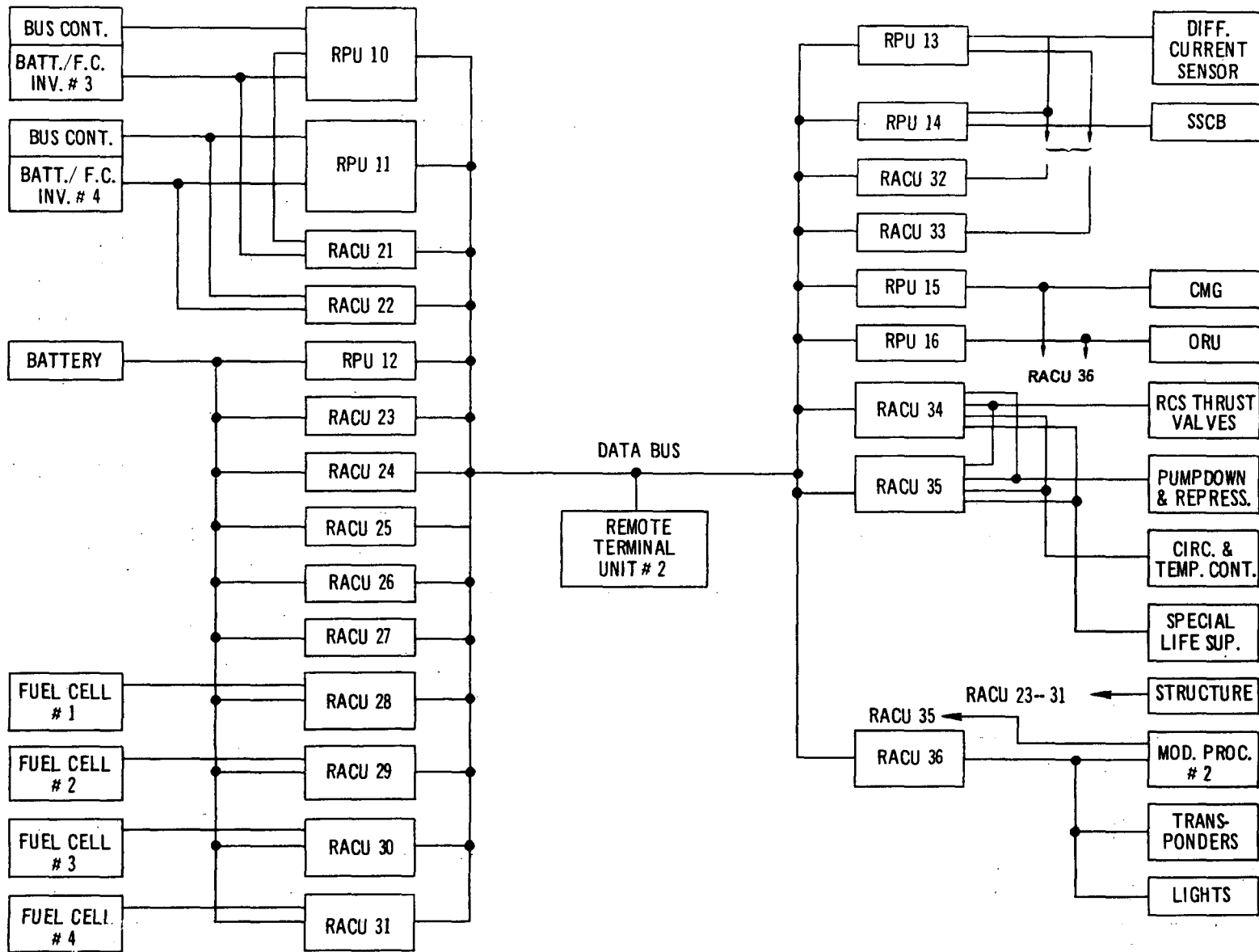


Figure 4-2. Core Module Hookup-Baseline

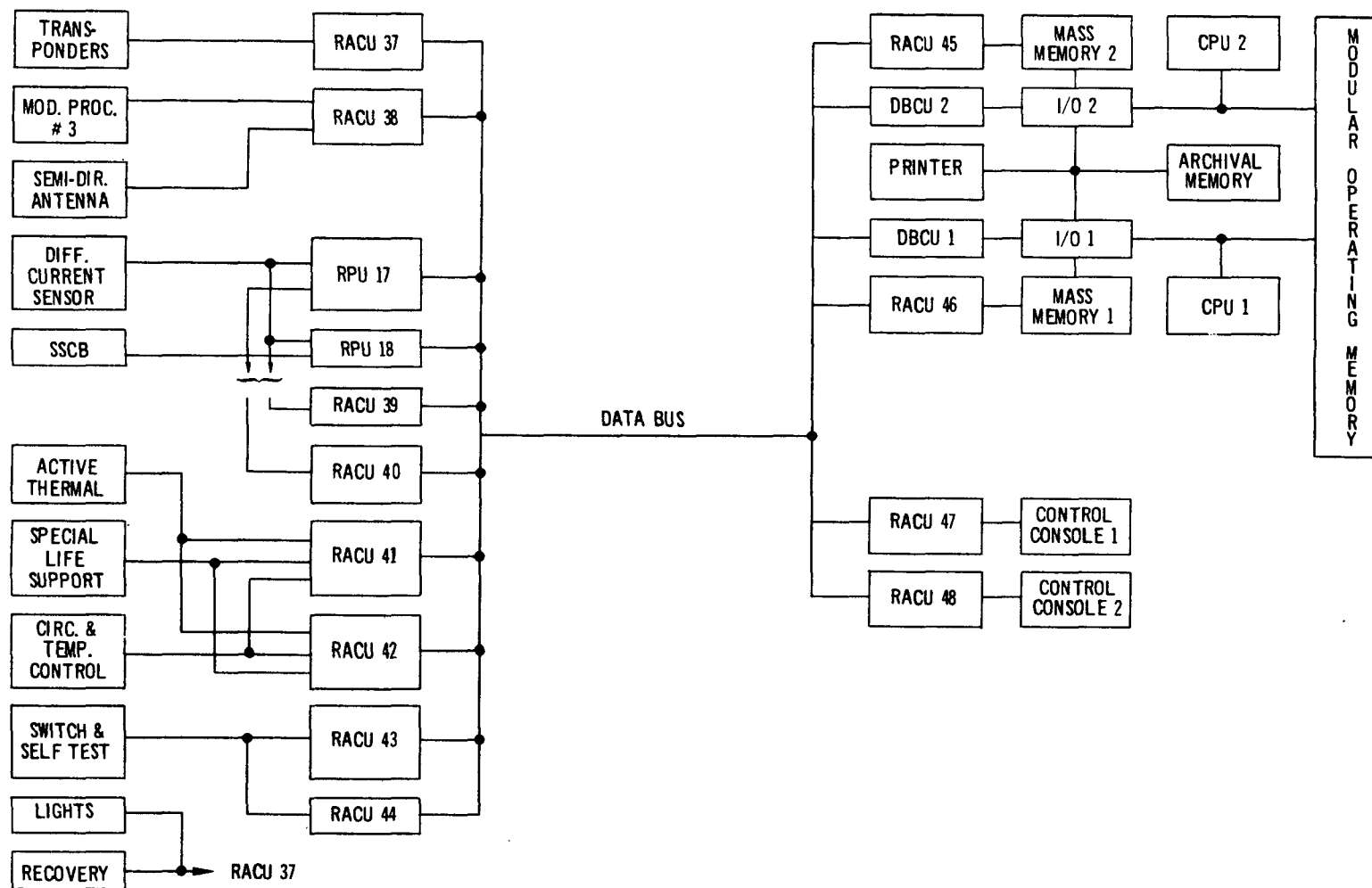


Figure 4-3.. Station Module No. 1 Hookup-Baseline

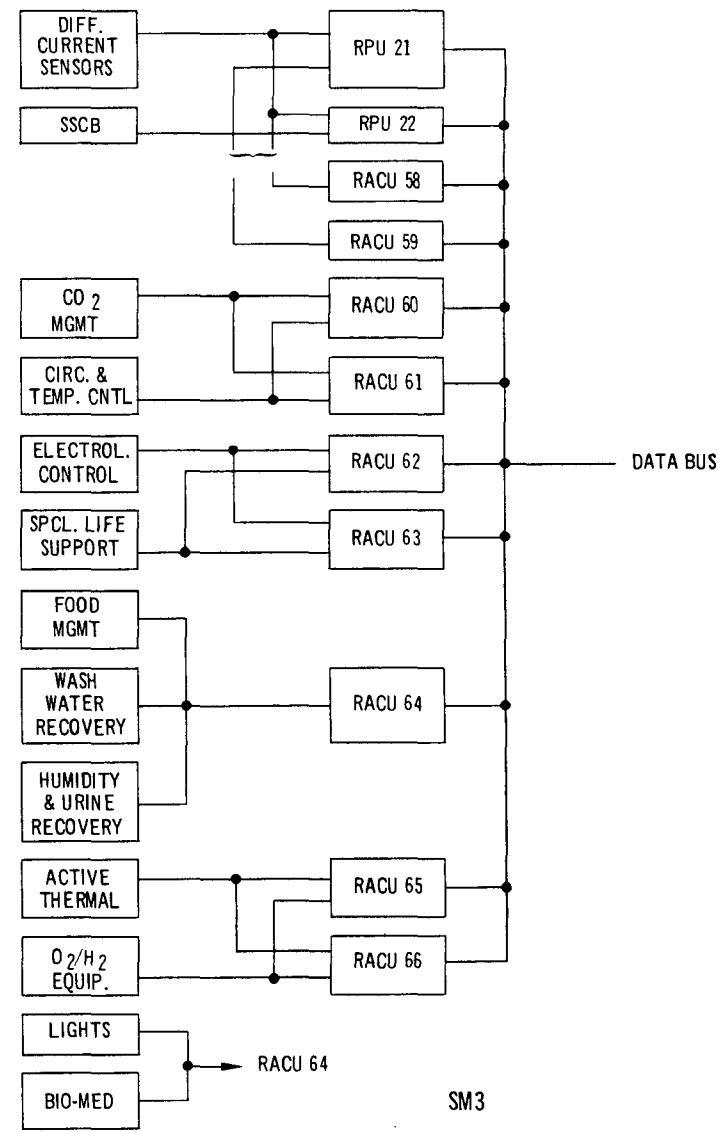
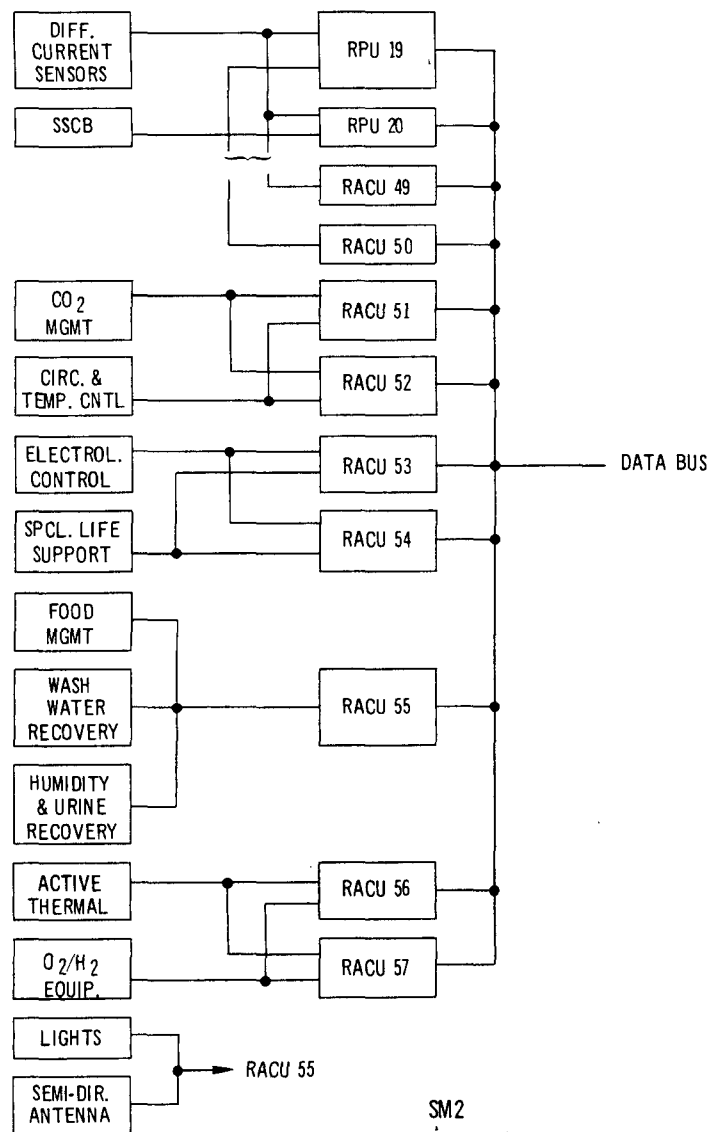


Figure 4-4. Station Modules Nos. 2 and 3 Hookup Baseline

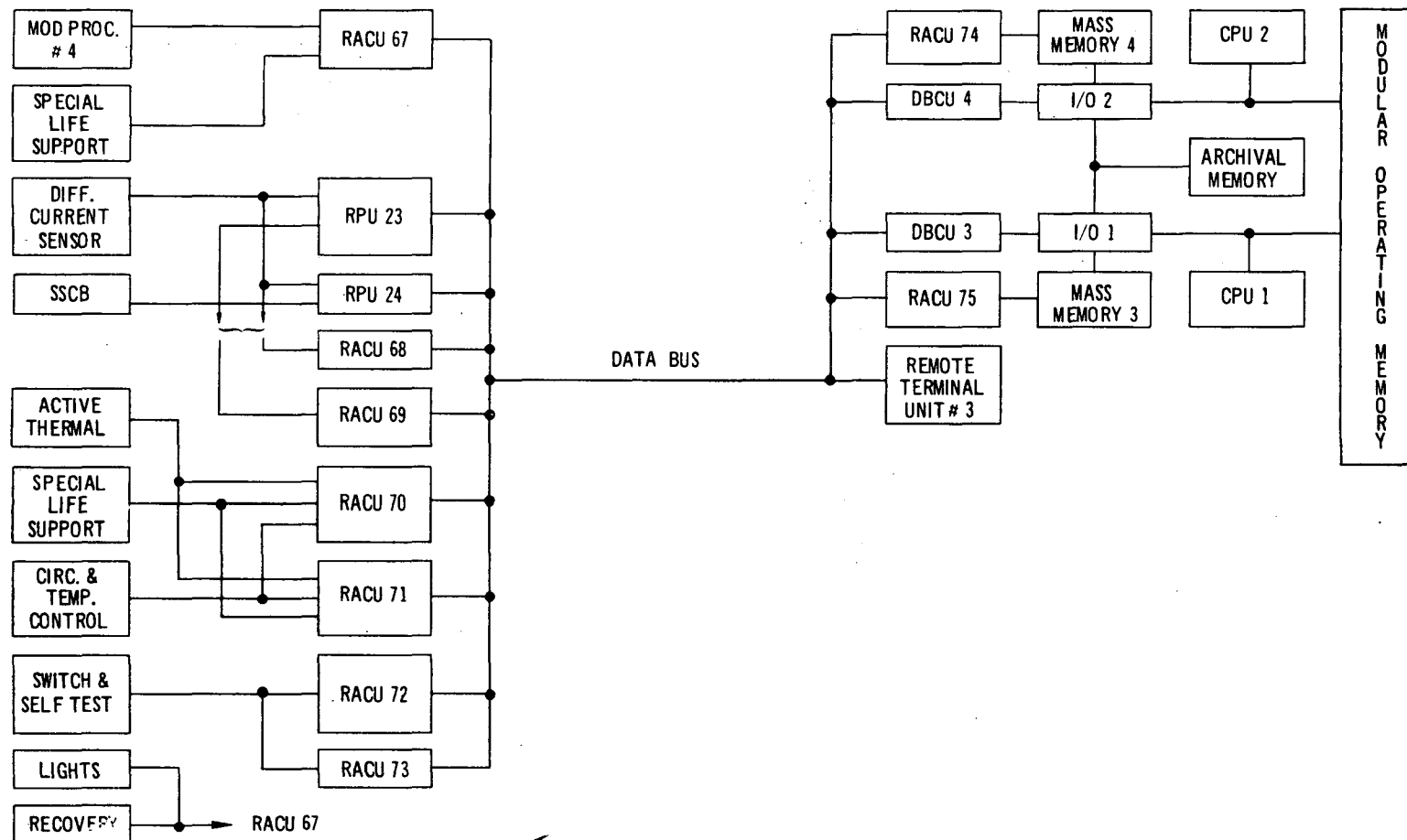
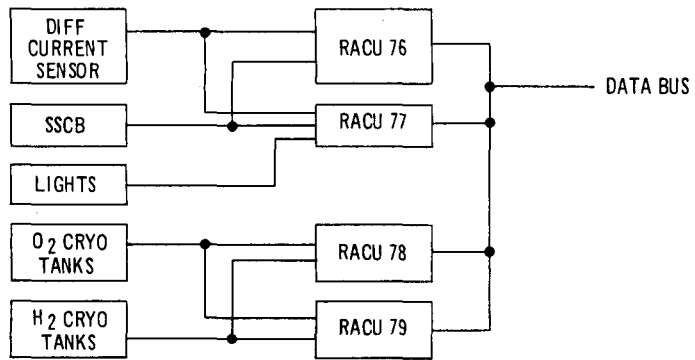
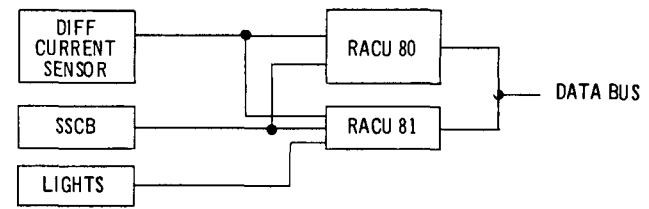


Figure 4-5. Station Module No. 4 Hookup-Baseline



CARGO 1



CARGO 2

Figure 4-6. Cargo Modules Nos. 1 & 2 Hookup-Baseline

Table 4-1. Preliminary DPA Interconnection Notes

Basis - The initial MSS Phase B vehicle failure criteria:

- a. Buildup Reliability Dictate
 - . Fail Operational
 - . Fail
- b. Manned Dictate
 - . Fail Operational
 - . Fail Degrade - 30 days survival, mission continuation
 - . Fail Emergency - 96 hours survival, mission continuation
 - . Fail
- c. Catastrophic Dictate
 - . Two pressure volume

Features

- a. Four power channels kept independent by providing RACU's and pre-processing on that basis.
- b. Within a module only dual redundancy is required since backup subsystem functions are provided in another pressure volume (A violation occurs for the solar array inverters 3 and 4. It's pre-processors are located in the power module along with those for SA inverters 1 and 2. Unless inter-module wiring is provided, loss of circuit breaker control affects all channels).
- c. RACU's are used for fault isolation. (The alternate approach of using pre-processors loaded with fault isolation routines and data from the central processor was not used in the sizing efforts. This latter approach affects operational and mass memory needs and needs to be evaluated further.)
- d. The data bus redundancy is defined to require such redundancy that any line pair (command-response) be able to connect to any two bus lines. Triple line pairs thus provided can also give another level for fail safe.
- e. Exact adherence to expressed RACU features was not observed for discrete-in-and-out. A review of these to determine if a digital word interface is better or whether 32 rather than 24 is more desirable is required. The maximum number of analog-in was held at 200, however.

FOLDOUT FRAME 1

FOLDOUT FRAME 2

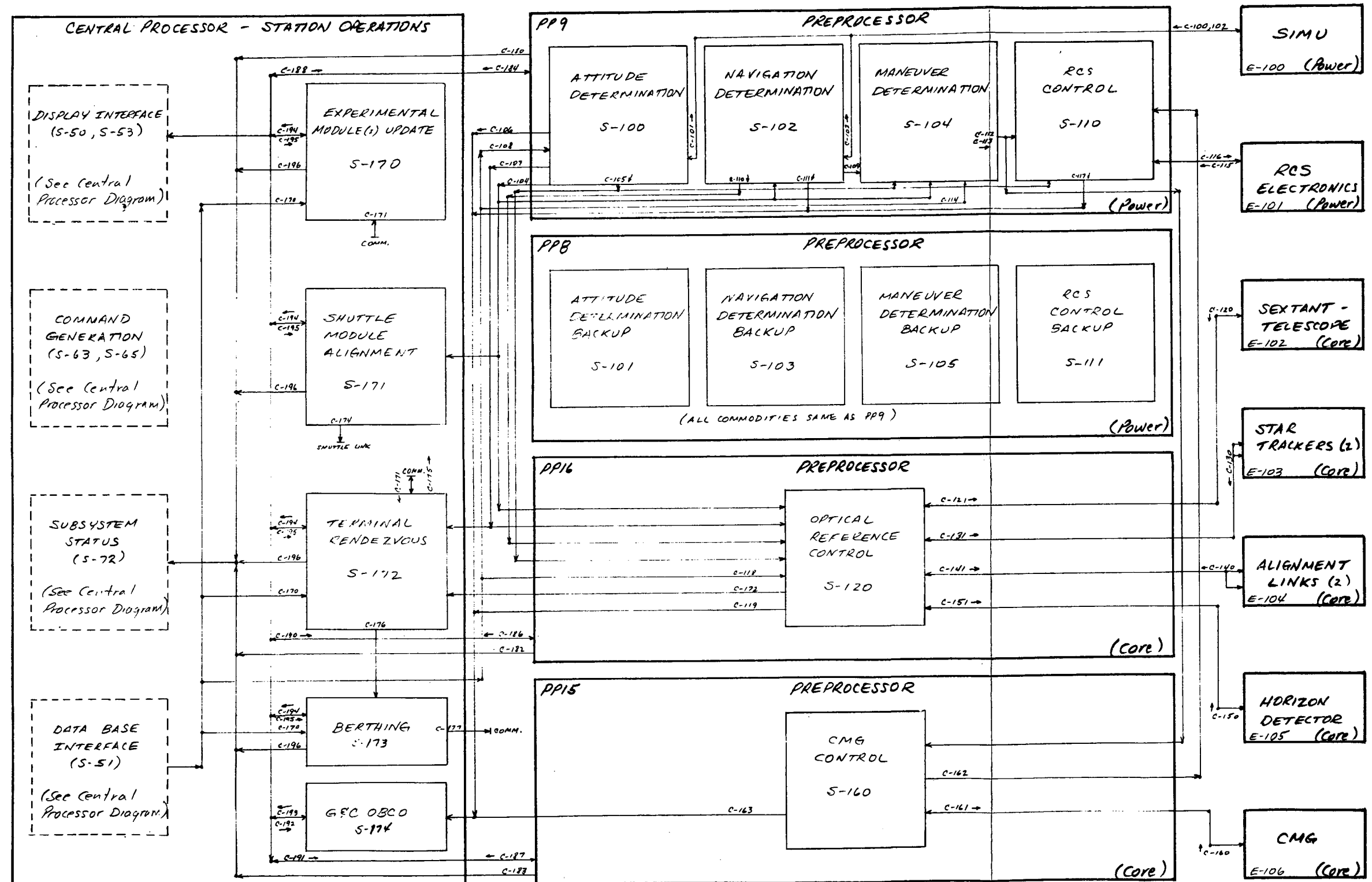


Figure 4-7. G&C Information Flow Diagram

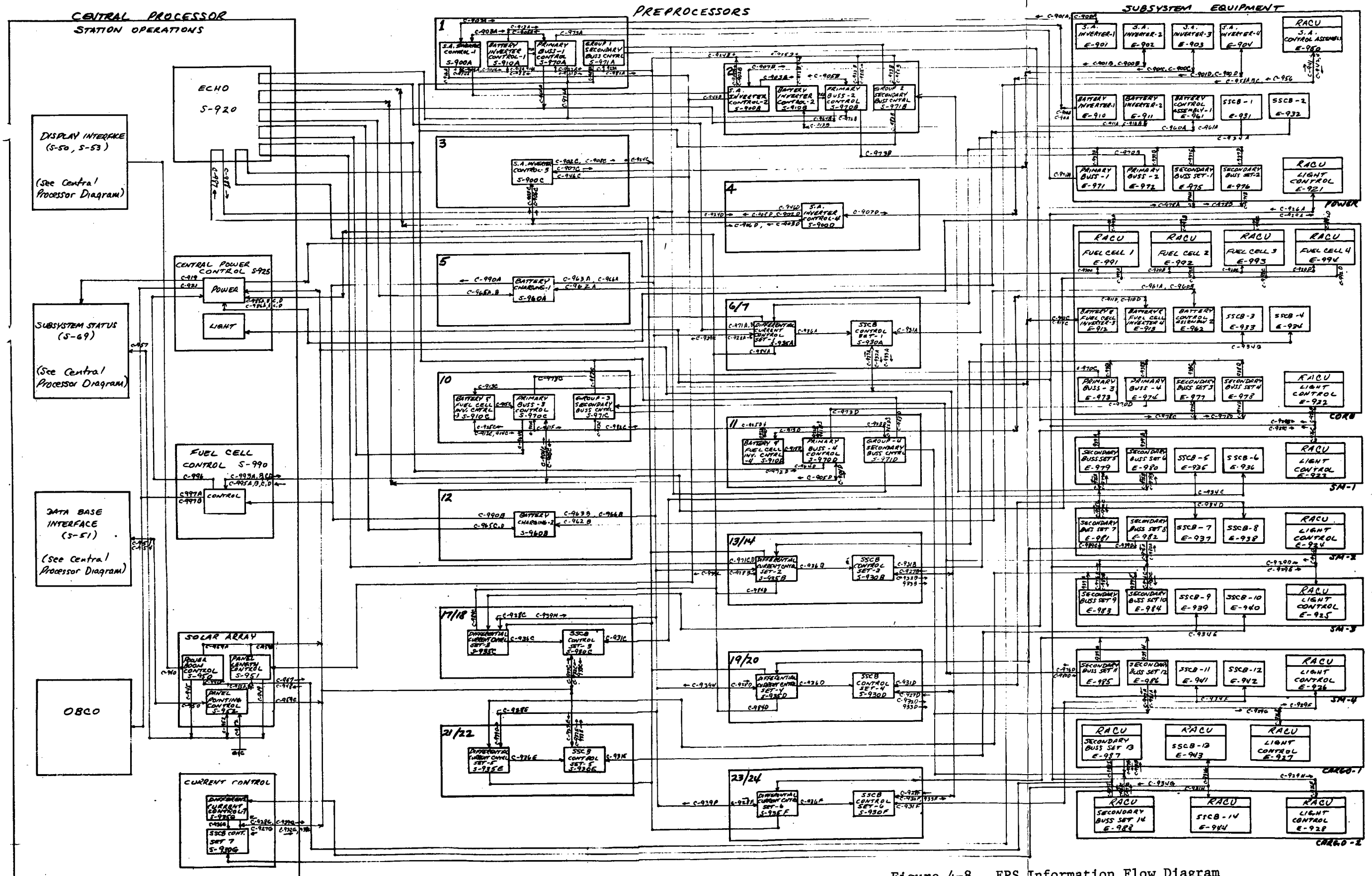


Figure 4-8. EPS Information Flow Diagram

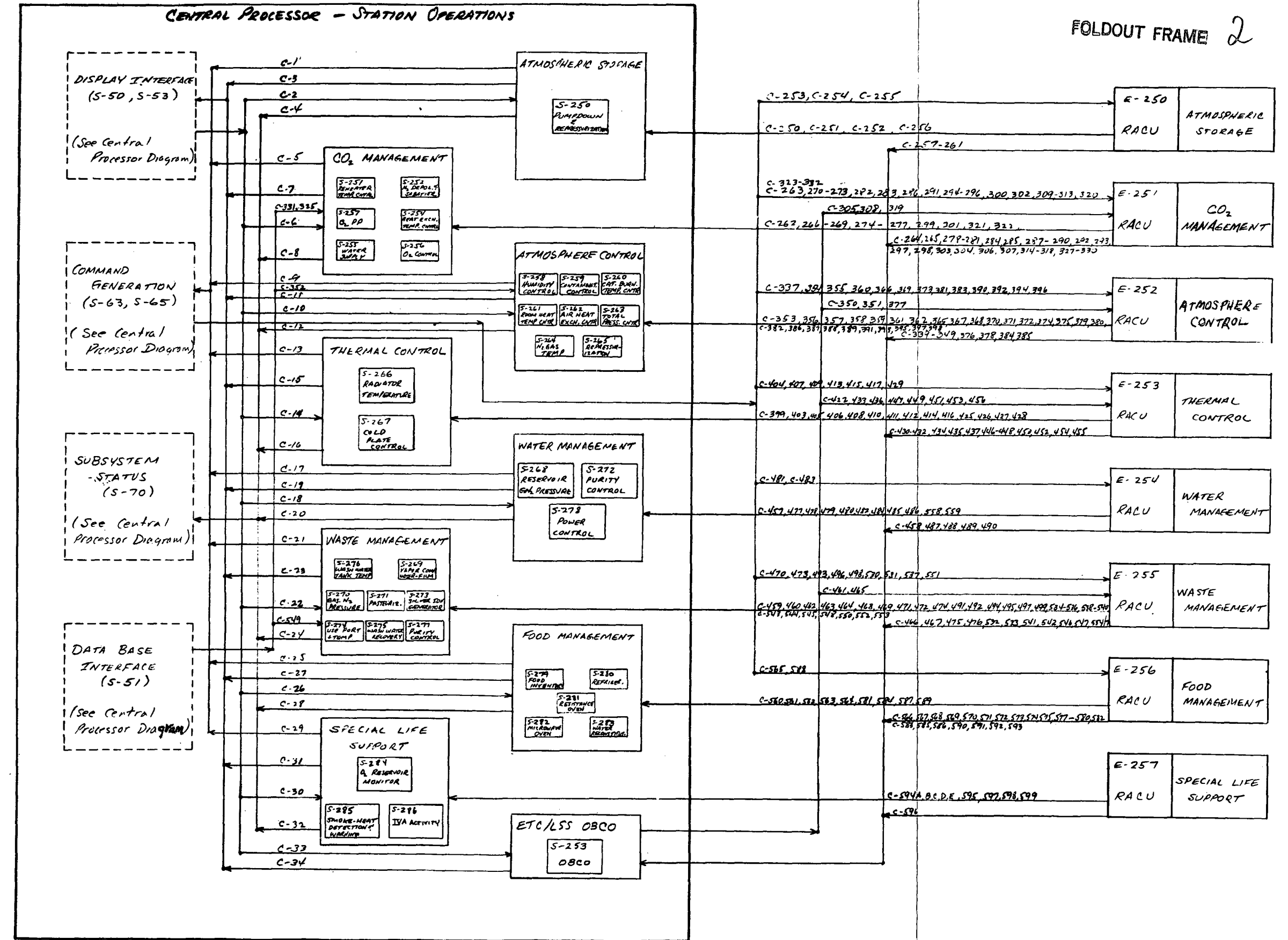
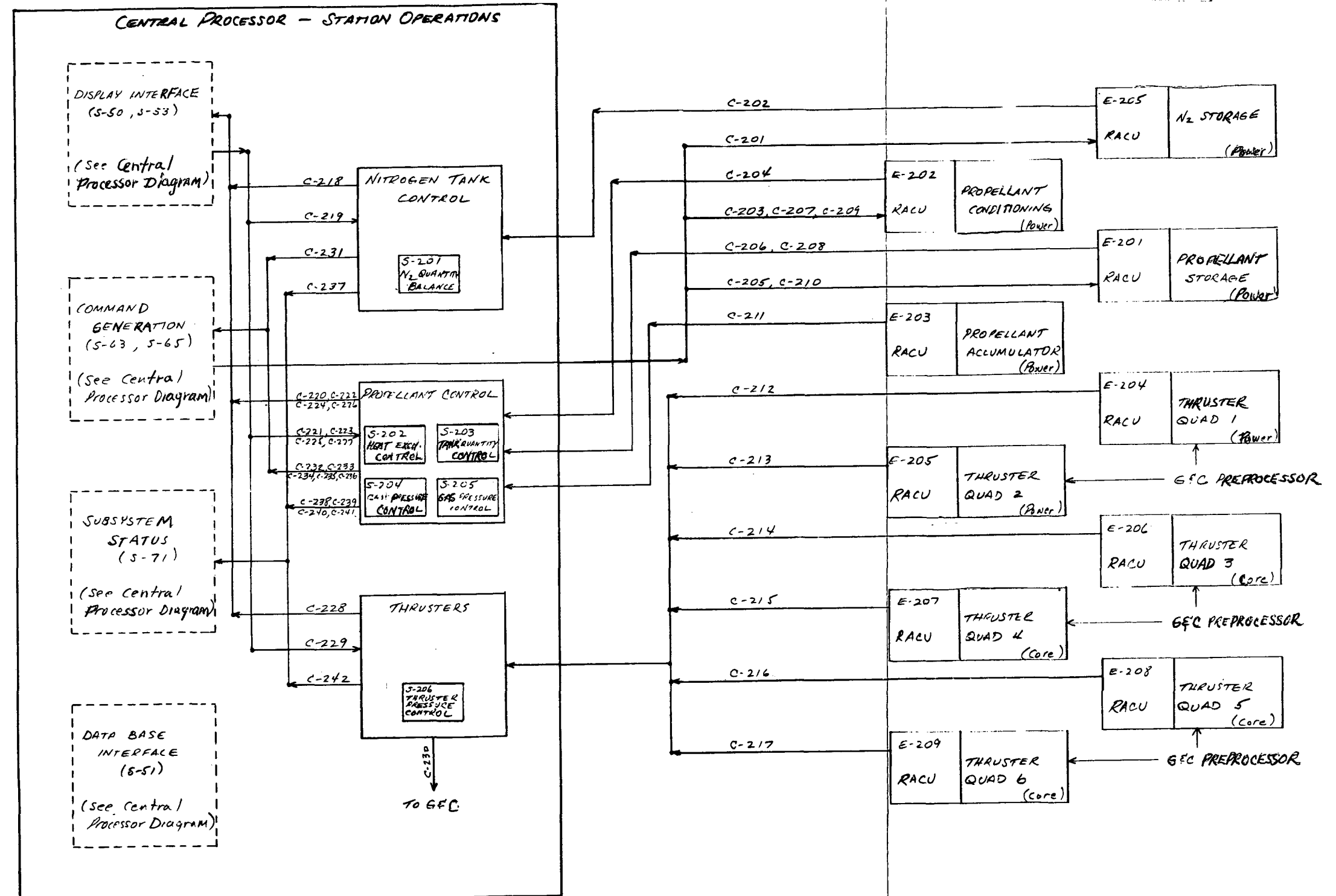


Figure 4-9. ECT/LSS Information Flow Diagram



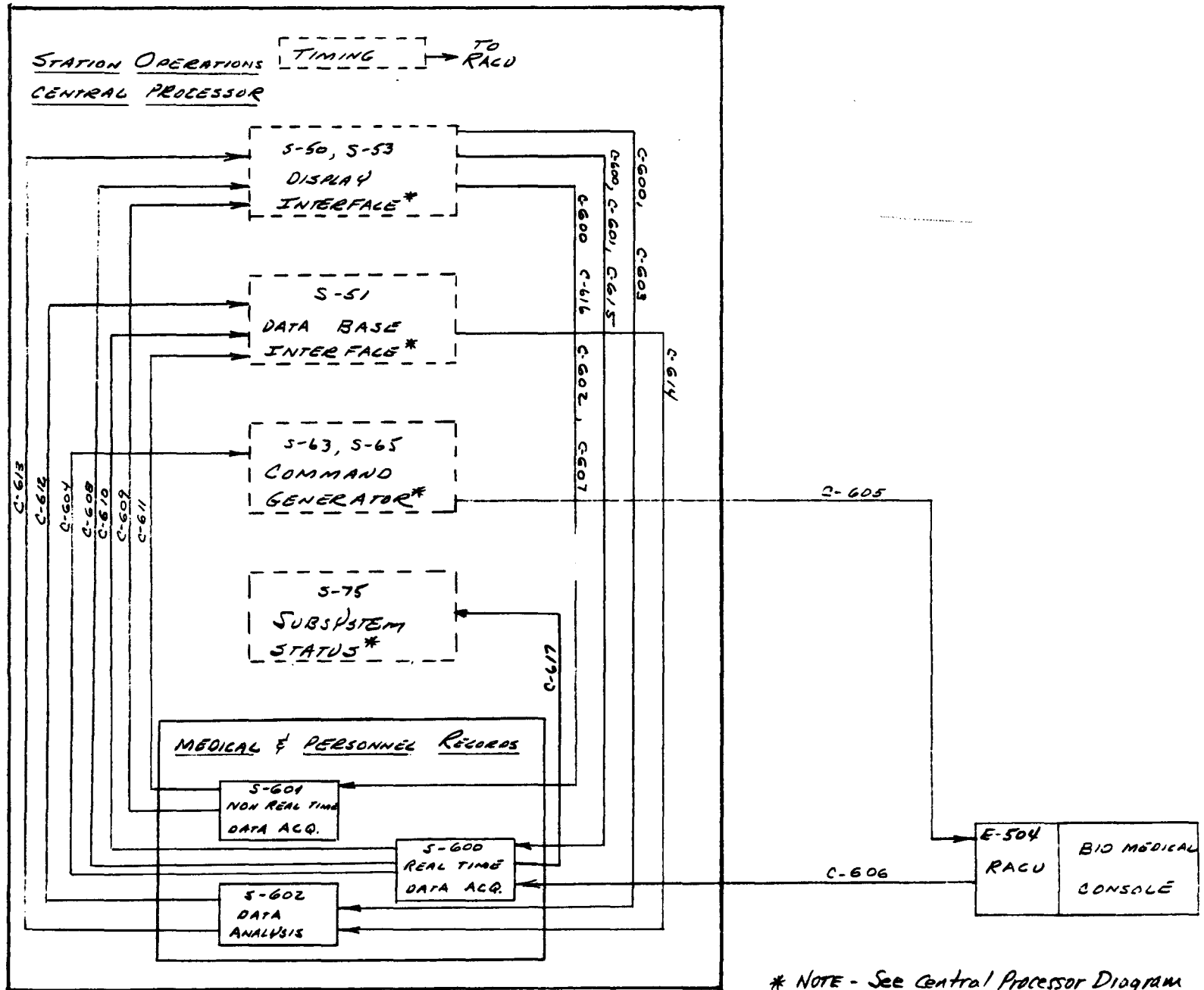
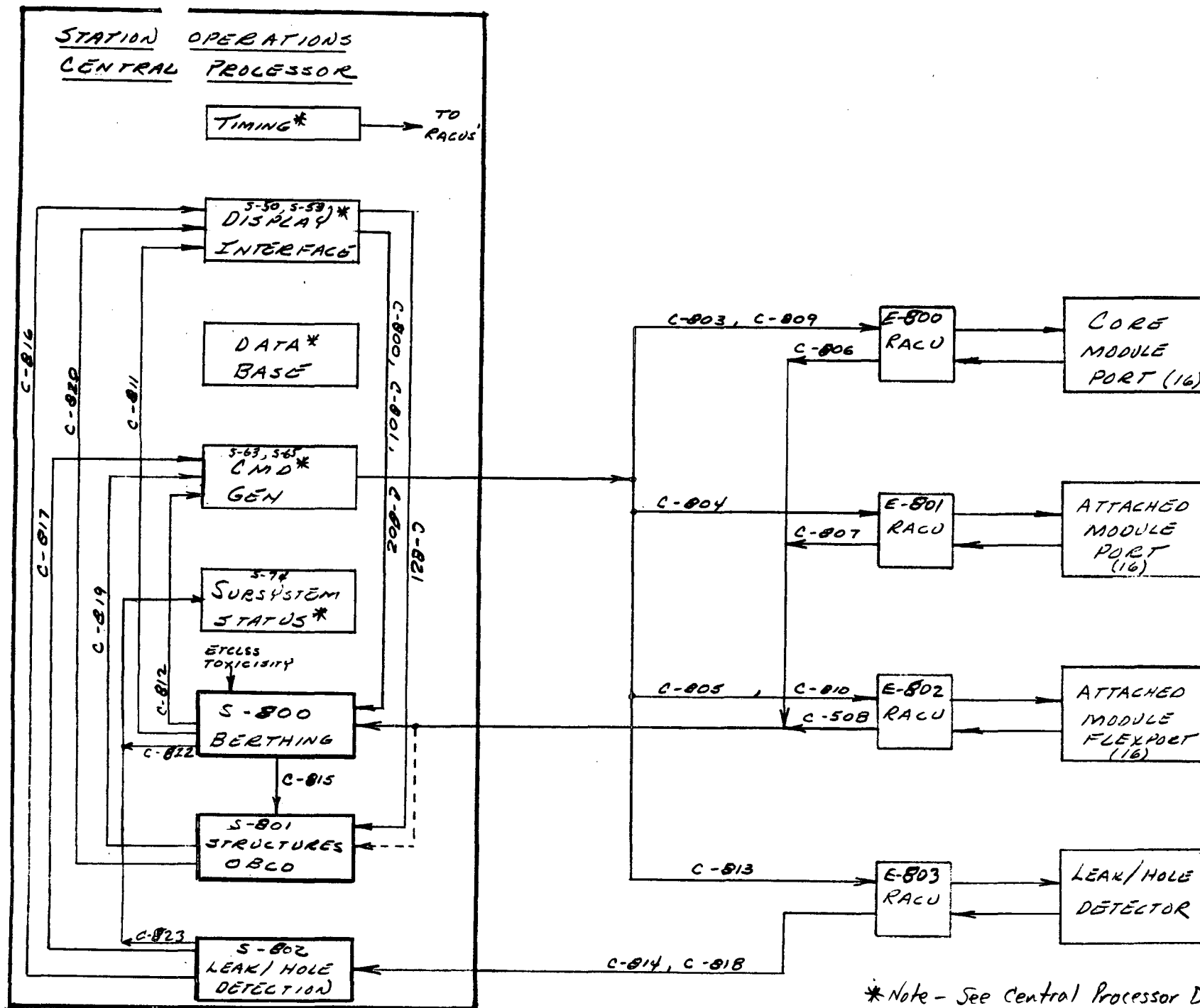


Figure 4-11. Crew Subsystem Information Flow Diagram



*Note - See Central Processor Diagram

Figure 4-12. Structures Subsystem Information Flow Diagram

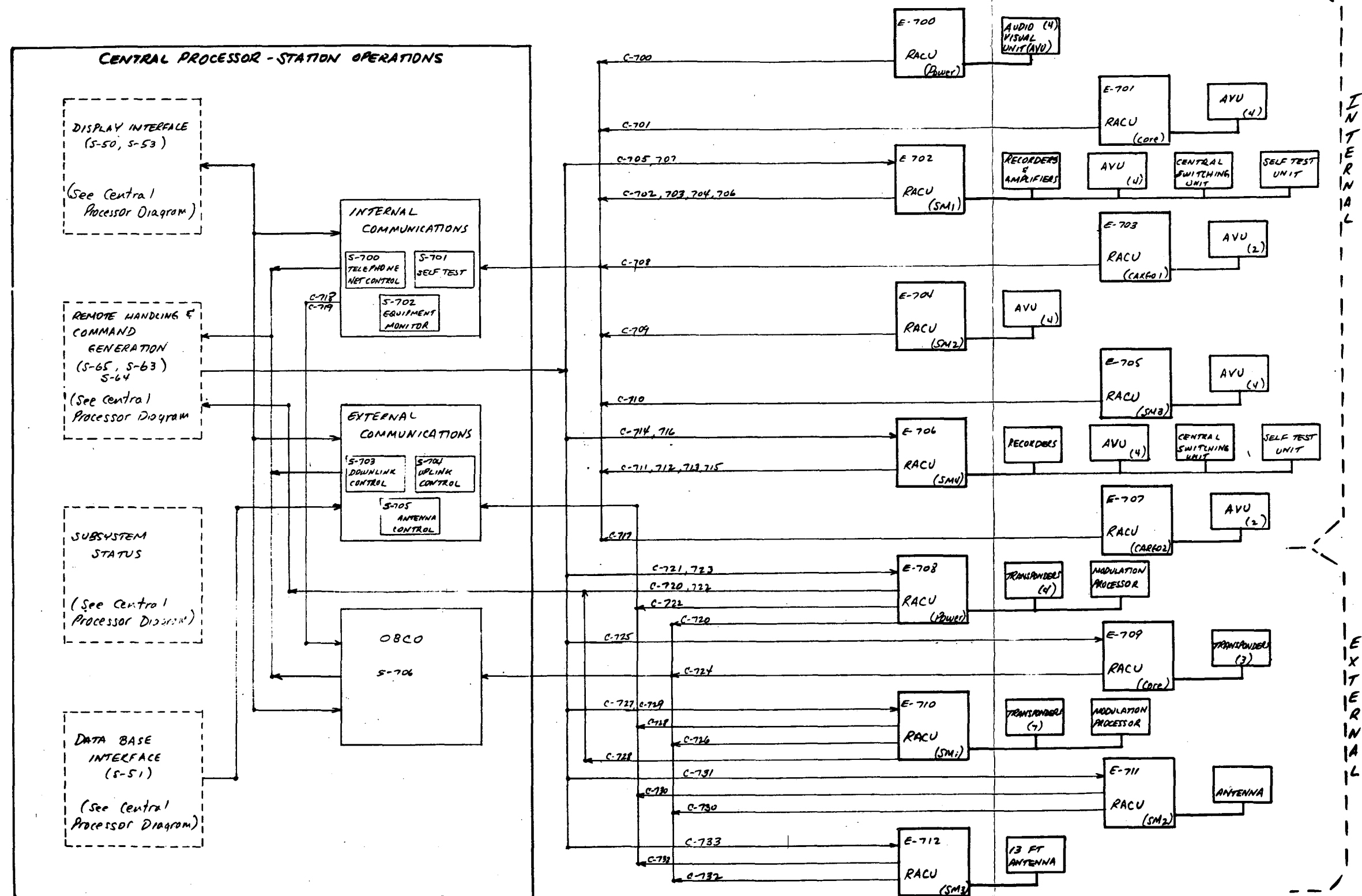


Figure 4-13. Communications Assembly Information Flow Diagram

FOLDOUT FRAME 1

FOLDOUT FRAME 2

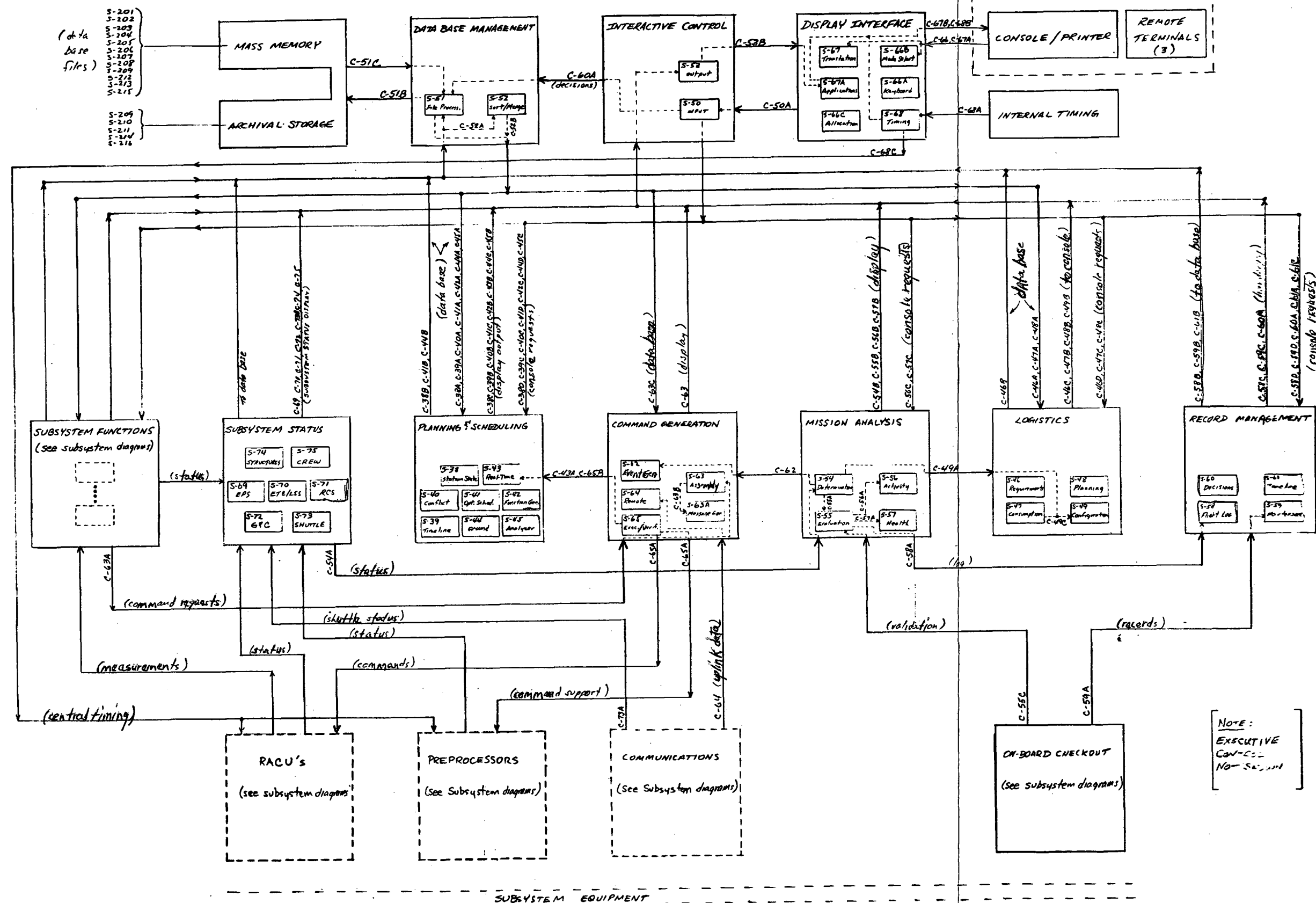


Figure 4-14. ISS Information Flow Diagram - Central Processor

peripheral devices are identified on the diagrams with numbers in the "C" series (for commodity). Software and commodity characteristics were tabulated onto data sheets and are available for review upon request.

An example of how information and commodities flow between computer program elements and hardware subassemblies is shown in Figures 4-15 and 4-16. These diagrams are repeats of Figures 4-14 and 4-13, respectively, where the flow is indicated by heavy black lines. The objective in this example is to point the directional antenna, using the DPA and software. Commands are generated at the keyboard of the control console by the astronaut operator.

The keyboard command causes a transfer of commodity C-67A "Console Parameters" from the control console to the display interface routines S-66A, 66B, and S-67. S-67, in turn generates commodity C-50A "Interactive Input Parameters."

The Interactive Input Control routine S-50 is activated and passes the request to the appropriate subsystem function S-705. S-705 generates a command request commodity C-63A which activates S-63 and S-63A Command Assembly and message generation subroutines.

The command subroutines assemble the command "point antenna" and transfer the command to S-65, the Command Execution subroutine which causes commodity C-65A (Assembled Command List) to be transferred to the Remote Acquisition Control Unit (RACU) E-711, which issues a signal to torque the antenna to the requested position.

RACU E-711 also samples various antenna measurements such as the servo null voltage and sends the information back to the central processor (S-705) via commodity C-730 (Figure 4-16). S-53 is activated and sends commodity C-53B to the Display Interface routine S-67A. S-67A generates commodity C-67B which produces a display signal on the CRT. The communication subsystem status update and a data base update is also performed but is not discussed in this example.

1 September 1971

System Development Corporation
TM-4754/000/01

ISS INFORMATION FLOW - STATION OPERATIONS CENTRAL PROCESSOR

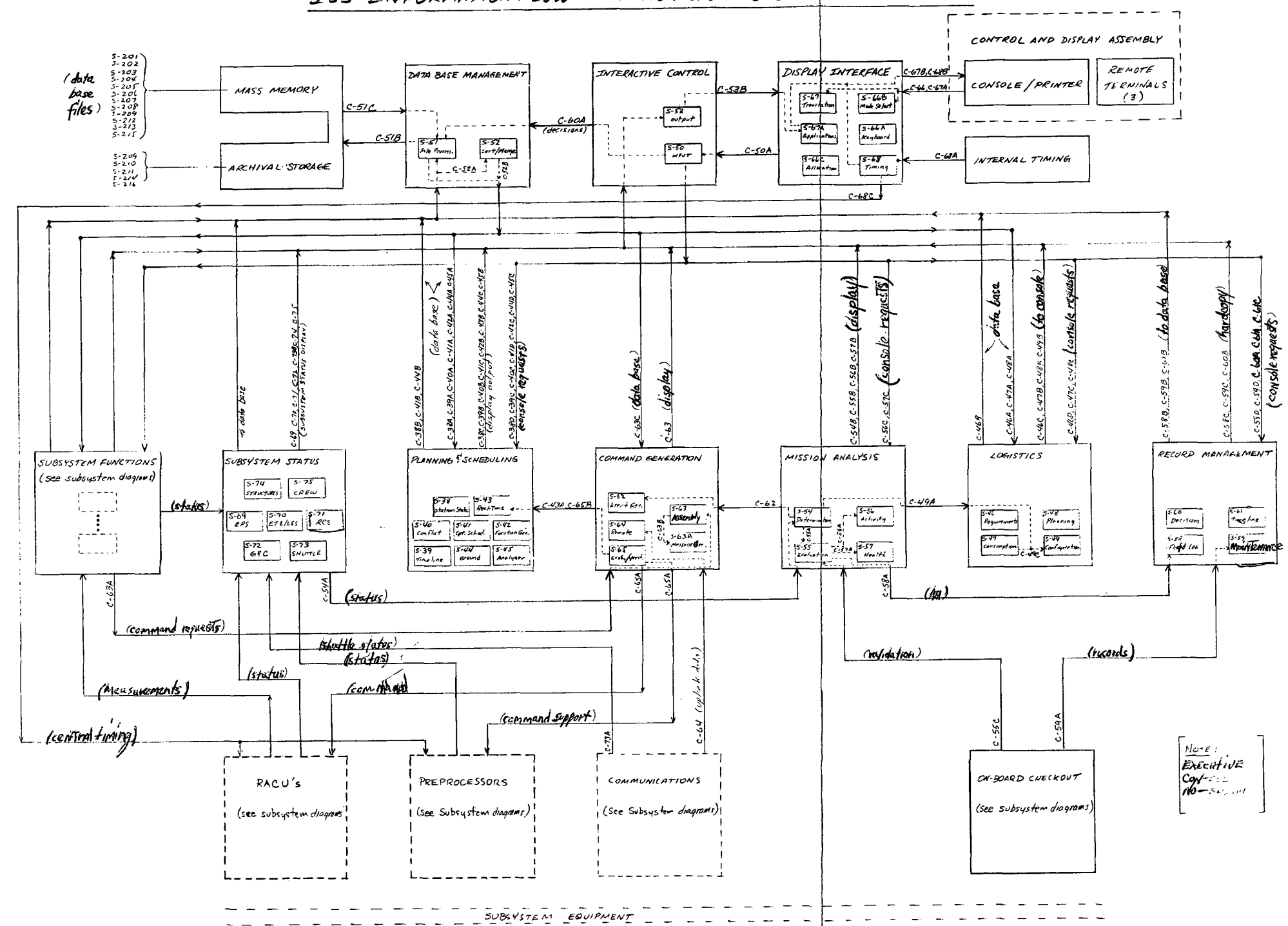


Figure 4-15. Information Flow Example (Antenna Pointing) - Central Processor Level

1 September 1971

System Development Corporation
TM-4754/000/01

COMMUNICATIONS - INFORMATION FLOW

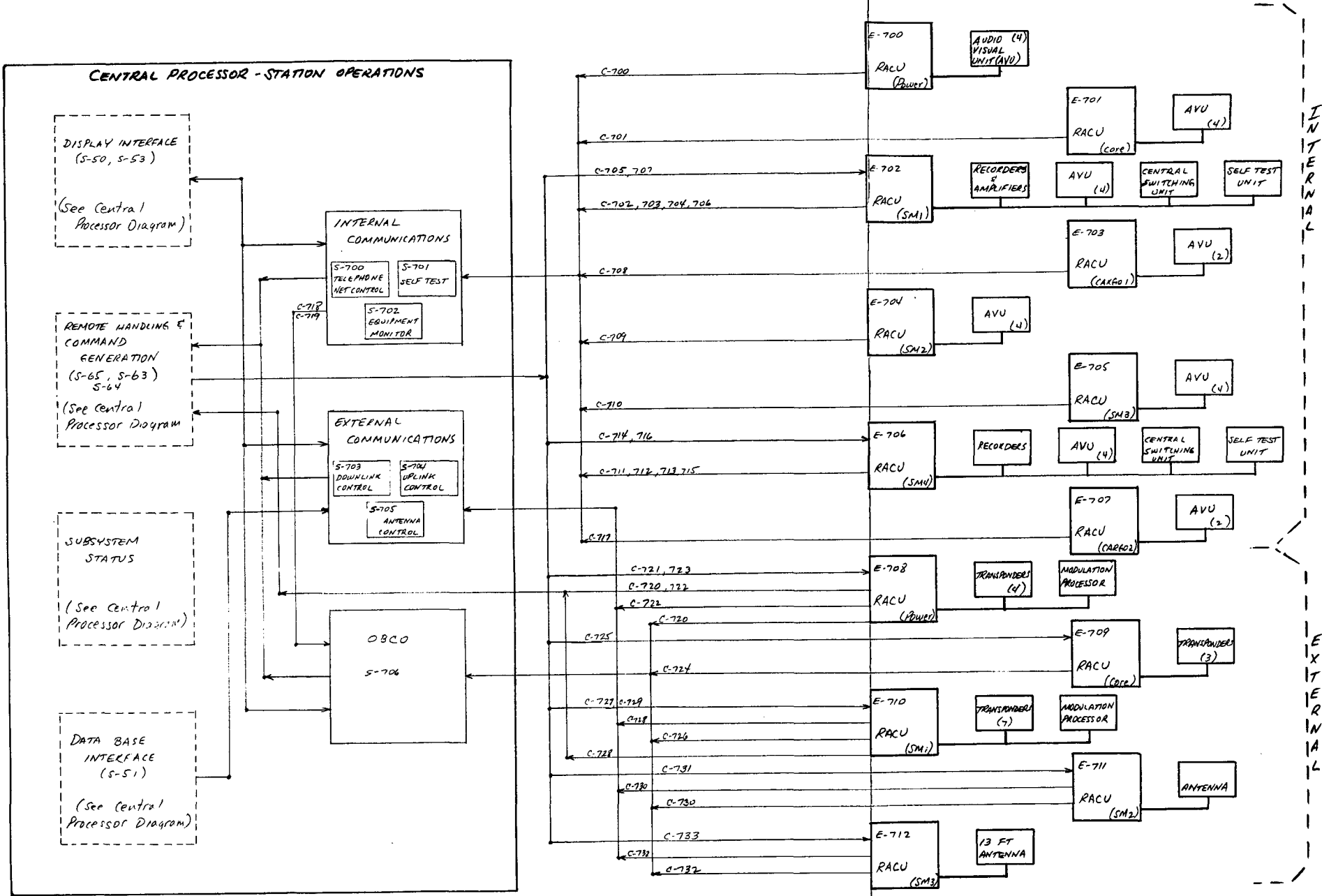


Figure 4-16. Information Flow Example (Antenna Point) - Subsystem Level

5. THROUGHPUT SIMULATION

5.1 INTRODUCTION

This section documents SDC's effort toward completing a throughput and authority analysis of the Modular Space Station's Information Subsystem Data Processing Assembly (DPA). The results obtained from the analyses are primarily predicated on the ADT configuration as defined in Section 3.

The objective was to provide information that would facilitate a selection of the final DPA configuration for the Modular Space Station; in particular, to provide information pertaining to DPA component performance that would yield a DPA configuration capable of accommodating imposed workloads within required response times.

Initially, the scope of the activities to be performed during this task was to determine the hierarchy of authority for the handling (processing) of data within the DPA. This hierarchy was to include the RACU's, the pre-processors and the central processors. Included within this task was to be the determination of the capabilities at each level of authority (such as conversion, switching command/response, processing, etc.) and the data transfer throughput to perform the function at the specified level for both response and command data.

This scope was redirected during the course of the task to the following set of study objectives, listed in the order of their priority of accomplishment.

1. Determine if the Advanced Development Task (ADT) DPA configuration will work. That is, has authority been properly assigned so that the performance of the baseline DPA will meet a pre-specified set of criteria.
2. If the ADT DPA does not work, modify on a parametric basis the operations of its elements until a workable DPA configuration is obtained.
3. When ADT DPA does work, determine the effects of transients (delays caused by failures and reconfigurations) upon the performance of the DPA.
4. Determine the effects of different types of configuration provisions on the performance of the DPA.
5. Investigate alternate DPA configurations. That is, investigate the effects of DPA performance generated by varying the ratio between centralized vs. decentralized processing allocations.



6. Investigate the effects of alternate workload/element characteristics upon the ADT DPA performance. For example, what are the consequences on DPA performance if fuel cells are used in lieu of batteries?

It was understood that the meeting of these objectives might be constrained by both the time and funds available. Under these constraints it was agreed that accomplishing a significant part of objectives 1 and 2 or 1 and 3 (i.e., provide some meaningful results that future MSS Advanced Study tasks can build upon) would constitute a successful accomplishment of this task.

The approach taken to assess the adequacy of the DPA has included a combination of analytical investigations which were verified by computer simulations. The following basic steps comprised SDC's approach to this task:

- a. Device Load Analysis

Each Remote Processing Unit (RPU) and Remote Acquisition Control Unit (RACU) was inspected to determine the amounts of data transferred and processed by associated functions (G&C, ECLSS, etc.). For each device, tabulations were made of the:

1. Response load on the data bus (R-BUS)
2. Command load on the data bus (C-BUS)
3. Transfer load (operating memory or mass memory-to-processor transfer).
4. Processing load (arithmetic processor load imposed by the memory-to-processor transfer).

These tabulations were performed for all sampling intervals envisioned for these functions (50 ms, 100 ms, etc.).

- b. Device Time-Line Summary

As an extension of step (a), summaries of the total transfer and processing loads for all devices were tabulated at all sampling frequencies. For example, if a RACU transmits data at 100 ms intervals for three subfunctions, the sum of these three loads would constitute the 100 ms load imposed by this device.

- c. Commutation Cycle Determination

The next step involved the determination of a commutation cycle rationale for sampling all of these devices at the required sampling rates. A fixed-cycle polling scheme was eventually selected as the most satisfactory means for retrieving required data via the data bus.

d. Commutation Cycle Slot Allocation

Once a rationale had been developed, the results of steps (a) and (b) above were used to determine a means of allocating devices to particular slots of the polling cycle. Particular attention was given to the impact on the Arithmetic Unit (AU) of the Station Operations Central Processor (CP), since the efficient use of this processor is a reliable indicator of overall CP performance.

As a related portion of this slot allocation effort, estimates were made of the adequacy of proposed arithmetic processor speeds to handle expected workloads.

e. Computer Simulation of ADT Configuration

To confirm the results of the numerical investigations of the preceding steps and to investigate the effects of the interactions of CP processing units and related elements, a computer model was constructed and simulation runs were performed. Several sets of statistics were accumulated, and the adequacy of the proposed ADT configuration was verified.

5.1.1 DPA Configuration and Performance Assumptions

At the outset of the throughput simulation the preliminary baseline DPA configuration (sometimes called the ADT configuration) had been defined (as described in Section 3). However, the results of the MSS Phase B studies in refining the DPA requirements were not then available (see paragraph 2.2). Hence, certain assumptions were made in regard to the DPA performance characteristics. In particular, it was assumed that the station operations central multiprocessor contains two I/O processors, two arithmetic units (AU), and two modular operating memories. In view of the fact that the requirements tabulated in paragraph 2.1.7 include experiments as well as station operations, the assumptions detailed below were deemed sufficient for consideration of station operations only:

Central Processor Assumptions

1. In regard to accomplishment of objective 1, (i.e., to verify that the DPA configuration can work), assume a single computer configuration within one CP multiprocessor. That is, only one of each type of element (I/O, AU, OM, MM, AM, RACU, DBCU) will be sufficient to verify concepts via simulation modeling methods.*
2. The arithmetic unit operates at 0.75 MAPS, while the I/O unit operates at 300K words (32 bits per word) per second.

* That is, each AU-I/O unit combination will generally be performing (or be capable of performing) the same operations. Therefore, the conceptual adequacy of the DPA configuration can be adequately proven by simulating one pair of these processors.

3. The executive governing the operations of the CP is assumed to be simpleminded; e.g., no paging.
4. The DBCU is assumed to be transparent to all incoming and outgoing signals.
5. The station operations central processor operating memory consists of at least four modules. Each module contains 32K-32 bit words of memory. Both the AU and I/O are connected to each module. A given memory module can only service one (AU or I/O) processor at a time.

Operating Memory Operational Characteristics:

- o Random Access
 - o 750 nanosecond cycle time
1. Mass memory characteristics per SD70-159-3*, where appropriate. (Note that tape recorder is not a normal on-line access device).
 2. Archival memory characteristics per SD70-159-3*, where appropriate.

RACU

1. Input/output rate - 300K words/second (must match bus transmission rate).
2. Standardized throughout the station.
3. Samples subsystem measurements and has measurements available for dump to the CP upon command from the CP.
4. Provides fault isolation capability for subsystem loops through the central processor, which are controlled and monitored by RPU's.
5. Monitors and controls subsystem loops through the CP which are assigned to the CP.
6. Has a 4K byte (8 bit) memory of which 600 bytes are overhead.

RPU

1. Monitors and controls subsystem loops in which it operates.
2. Provides status of each subsystem loop to the CP on a cyclic basis.
3. Detects subsystem loop failure, switches in redundant subsystem loop, and notifies CP which subsystem loop has failed.

*"Solar Powered Space Station Preliminary Design"

4. Has: a. 6K word memory of which 900 words are overhead
b. Operates at 500K equivalent adds/second
c. I/O rate: 300K words/second (must match bus transmission rate).

5.2 TIME SLOT ANALYSIS

The operational doctrine assumed to be in effect for the data bus is that of polling. Polling control is assumed to be a function of the I/O unit and the polling schedule is assumed to be on a "fixed" time basis per device. The polling schedule assumed in the authority and throughput analysis is predicated on dividing a second into 250 slots of 4 ms each.

The 250 slot assumption was obtained by considering that the DPA has, in accordance with step (B) of paragraph 5.1, the following device sampling rate requirements:

<u>Number of Devices</u>		<u>Shortest Sampling Interval Per Device</u>		<u>Required Samplings Per Second at Highest Rate</u>
5	@	50 ms	=	100*
9		100 ms		90
44		1 sec		44
2		10 sec		0.20
13		60 sec		0.22
Total 73			Total	235

* e.g., $\frac{1 \text{ sec}}{0.05 \text{ sec/sample} \times 5 \text{ devices}} = 100 \text{ samplings per second.}$

Approximately 235 slots are thus required if each of the 73 devices is to have its own time slot to report to the CP at its highest required frequency. Allowing an additional 15 slots to handle contingencies such as OBCO-Fault Isolation or Real-Time Bio-Medical inputs brings the total slots required per second to 250. For these 250 4-millisecond slots, the polling schedule is essentially periodic with a period of 25 slots. The slot allocation for one 25 slot period is as follows:

Slot Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
50 ms devices (plus 2 > 1 second)	A		B			C		D			E			A		B		/	C			D		/	
100 ms devices (plus 4 @ 1 second)		F		G	H		I		J	K		L	M		N	1				2		3			4

→ | | ← 4 ms

where A, B, C, D and E represent the slots for polling 5 devices that have sample rates of 50 ms.

F, G, H, I, J, K, L, M and N represent the slots for polling 9 devices that have maximum sample rates of 100 ms.

1, 2, 3 and 4 represent the slots for polling 40 devices, four per period, that have a sampling rate of 1 second.

✓ represents the slots for polling those devices with a sampling period of one second or more.

5.3 ANALYTICAL RESULTS

Based on the tabulations described in the preceding sections, a numerical analysis and tabulation of expected arithmetic unit utilization was performed. Particular attention has been given to the use of this processor, since its operations are the heart of the CP. Thus, if it can be shown that an arithmetic unit can satisfactorily accommodate the software workload imposed by DPA devices (at given bus transfer rates), high confidence may be realized in the functioning of the entire DPA as an efficient system.

The arithmetic unit analysis proceeded as follows:

For each slot, a specific RPU, RTU (Remote Terminal Unit) or RACU will be sampled. The expected load placed on the arithmetic unit by each device in its time slot has been estimated by totaling the expected operations from the following formulas. Note that each of these six equations may not apply to every device. For example, RACU 16 (employed for data on RCS thrust valves and $O_2H_2N_2$ tanks) does not use equations B, C or F (output call message, transfer of data to OM, and output processing, respectively), but it does utilize processes A, D and E. Thus, the load tabulated for each device consists of the totals of the applicable portions of the following equations:

- | | |
|--|---|
| A: Input processing and interpretation | = input transfer load + 426 operations for pre-processing, interpretation, post-processing, and executive control (10, 391, 10 and 15 operations, respectively) |
| B: Output call message | = 37 operations for executive control, I/O processing, transfer, and access (15, 20, 1 and 1 operation, respectively) |
| C: Transfer data to OM for storage | = OM data transfer load and 36 operations for executive control, I/O processing and access |
| D: Transfer program in from OM | = OM program transfer load and 36 operations for executive control, I/O processing and access |

- E: Perform applications processing = number of operations required + 15 words for executive control
- F: Output processing = number of output words + 20 words for I/O processing

As indicated earlier, a single device may require transmission (sampling) at several different rates. Thus, these computations were carried out for each sample rate of each device. The worst-case load would then be the maximum possible load imposed on the arithmetic unit when all samples occur simultaneously. For example, RACU 37 transfers data at 1 sec, 10 sec, and 100 sec intervals: at every 100 seconds, this total maximum simultaneous load can be expected.

This worst-case approach was used in constructing the cycle loading postulated in Table 5-1, "Arithmetic Unit Utilization Analysis." Each slot entry in this table indicates the hand-calculated operations required to support the DPA devices, grouped into the commutation cycle sequence shown in paragraph 5.2.

Table 5-1 contains the following data:

1. The column labeled "Slot" (No. and Ident.) identifies the twenty-five slots of a slot group.
2. The column labeled "Slot Group 1" shows the device assigned to each slot in the group and the number of operations required to process the workload of that device.
3. Columns labeled "Slot Group 2, 3 and 4" are similar to that of Slot Group 1. For a given slot identification, the operations differential between Slot Group 1 and Slot Group 2 reflect the difference in processing the worst-case load and the periodic load.
4. The column labeled "Slot Groups 5-10" shows the operations required for processing the workload from the devices that would be in the numbered or checked slots if all ten slot-groups were tabulated (does not represent slot assignments).
5. The entry labeled "Periodic" in the "Slot Groups 5-10" column shows the operations required to process the periodic workloads for those devices assigned to lettered slots in Slot Group 2, summed over the last six slot groups.
6. The row labeled "Overhead" accounts for the operations required to perform the CP background loadings and to output the words (commands, displays and printer) assumed during each slot.

Table 5-1. Arithmetic Unit Utilization Analysis

Slot		Slot Group 1		Slot Group 2		Slot Group 3		Slot Group 4		Slot Groups 5-10	
No.	Ident.	Device	Ops	Device	Ops	Device	Ops	Device	Ops	Device	Ops
1	A	RACU 35	3012	RACU 35	769	Same	Same	Same	Same	RACUS 1,2,3,4	4416
2	F	64	7435	64	562					17	1878
3	B	63	769	63	769					20	1664
4	G	RPU 15	990	RPU 15	0					53	3631
5	H	9	3022	RPU 9	0					56	1362
6	C	RACU 42	1993	RACU 42	1057					65	1362
7	I	16	1638	16	1638					70	1362
8	D	54	769	54	769					78	1417
9	J	34	2799	34	2799					43,72	51626
10	K	76	853	76	716						
11	E	71	1739	71	769					RPU 1	684
12	L	77	2403	77	1264					2	668
13	M	80	853	80	716					3,4	1028
14	A	35	769	35	769					5	624
15	N	81	2404	81	1264	✓	✓	✓	✓	6,7	1022
16	B	63	769	63	769	Same	Same	Same	Same	10	616
17	1	RPU 16	21290	79	3447	RACU 28	2594	RACU 66	2158	11	621
18	✓	RTU 1	4122	19	3781	OBCO-FI	2974	18	1398	12	624
19	C	RACU 42	1057	42	1057	Same	Same	Same	Same	13,14	1008
20	2	55	7732	67	3809	RACU 29	2594	RACU 51	2169	17,18	1054
21	D	54	769	54	769	Same	Same	Same	Same	19,20,21,22	1976
22	3	37	5645	31	3002	RACU 30	2594	RACU 60	2169	23,24	1054
23	✓	47&48	67531	52	3616	61	3616	41	1362		
24	E	71	769	71	769	Same	Same	Same	Same	Periodic (X6)	103350
25	4	38	6471	53	3807	RACU 57	2158	RACU 36	2587	Executive	55332
Overhead			10000		10000		10000		10000	Overhead (X6)	60000
Slot Gp. Totals			157603		62548		43755		39068	• • •	298379
Cum. Req. Ops Et*			157603		220151		263906		302974	• • •	601353
Cum. Avail. Ops ET*			75000		150000		225000		300000	• • •	750000

*Et = cumulative required operations

ET = cumulative available operations at a speed of 750K EAPS

Note that a preponderance of processor requirements have been placed into Slot Group 1; i.e., no attempt has been made to evenly distribute processor requirements over a full 1 second period. Thus, in the first 100 ms, it is noted that over twice as many operations (157,603) are required than those possible at a processor rate of 75,000 operations per 100 ms. However, as this table indicates, all processing can be completed within a one second period at a duty cycle of approximately 80 percent:

$$\frac{601,353 \text{ required operations per second}}{750,000 \text{ available operations per second}} \quad 80 \text{ percent}$$

Therefore, if loads are distributed more equitably, or if processing backlogs can be tolerated, a 750,000 ops per second arithmetic processor appears adequate to handle anticipated station operations loads. Moreover, if additional processing is made available from the duplexed arithmetic unit, the speed requirements can be reduced.

Note also that within the first slot group of Table 5-1, RACU 35 (Slot "A") has a 3012 EAPS entry in slot 1, while only 769 EAPS appear in slot 14 and subsequent slots. This is an extension of the worst-case grouping to consolidate as much processing as possible into the front end of an individual slot group; i.e., it was assumed that the 50 ms and 1 minute sample loads all occurred in the first slot of the first slot group. This assumption imposes an added constraint on the simulation modeling discussed later; that is, if the configuration can be shown to be satisfactory under these saturation conditions, greater confidence can be had in the workability of the postulated DPA.

5.4 DPA SIMULATION MODEL

This section describes the generation of a version of the simulation model used to assess the adequacy of the DPA configuration. This model, termed "IMSIM", has been tailored to produce meaningful results for this analysis, and used to verify the results of the numerical analyses. Summary results from the execution of simulation runs are presented in paragraph 5.5.

5.4.1 Simulation Model Characteristics

The capabilities incorporated into IMSIM have been oriented towards providing flexibility in representing computer system configurations and their workloads. Characteristics of equipment which are specified in the model by input parameters have been selected as those which could significantly impact the behavior of computers and communication links. Significance as used here, refers not only to the relevancy of characteristics, but also to the impact magnitude, considering the granularity of simulated time and space. Thus, characteristics such as weight and shape are considered irrelevant, as are functions completely external to the computer system. (Such functions may, however, be represented as response characteristics for the appropriate devices). Likewise, localized control signals such as interrupts have transmission times and data loads which are insignificant, even though

the implied functions may not be. For example, a signal to connect a device for transmission may be delayed until the device recognizes that signal, but the resulting connection can then be established virtually instantaneously.

The equipment simulation categories used in this model cover five basic types of equipment: memory units*, storage units*, computer processors, data transmission links, and a group called "devices" that includes all hardware not in the preceding four categories. Three additional categories are included by expanding the concept of equipment: data sets, system configuration specifications, and executive algorithms. The data sets provide an additional degree of freedom in loading storages and directing data transmission, while system configuration specifications identify methods of equipment interconnection. Executive algorithms, on the other hand, specify methods of directing software control over job and task execution (i.e., over DPA "users").

The "users" are represented in a hierarchical structure which permits workloads for the IMSIM to be organized and associated with conceptual capabilities of the DPA. The structure is shown in Figure 5-1. The "job" represents a complete function such as "plan flight path change", which may be broken down into tasks such as "prepare attitude change", "prepare spin-despin change", etc. The "tasks" are described in terms of the required routines, data blocks, and the messages to be transmitted over data links.

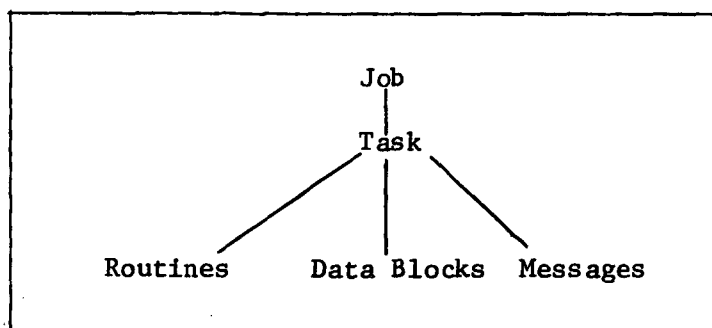


Figure 5-1. IMSIM Workload Hierarchy

The interdependence of tasks within a job may also be specified. This approach offers several advantages: subdivision of the job into tasks permits parallel processing for the job in a multiprocessor or multicomputer system; different processor requirements may be specified for various portions of the job; and different jobs may involve the same type of task. This latter feature permits prototype tasks to be defined only once, thereby avoiding redundant inputs.

*"Memory units" are used to simulate AU and I/O local memories, while "storage units" are used to simulate mass and operating memories.

The "message" has been chosen as the principal driver for the model. A message specifies a loading for data links and other equipment in communication, plus the amount of processing required for each transmission. This method of operation acknowledges the fact that a program may operate for varying lengths of time, dependent on the types and quantities of data to which the program is applied. Thus, routines and data blocks are essentially relegated to the status of space-takers (although their presence in local memory may be essential to performance of a task).

The IMSIM data base is structured to facilitate retrieval of information which is required for (or potentially useful to) the algorithms which represent executive control of the system. A centralized table concept is generally employed through which data can be prepared by transactions* in given parts of the model for use or regulation of transactions in other parts of the model. An example of such a table is the task table which contains the complete status of every extant task. Data which are solely for use in connection with a single transaction are retained by that transaction as transaction "parameters"; this association is preferable to global (in contrast to "local") representation, and is used whenever possible, because it simplifies the formulas for servicing transactions as these transactions are moved about in the logical block network.

The global/local classification of data is also well suited to preparation of outputs, since the purpose of simulation is to observe general characteristics of the model behavior and effects on statistically significant populations, rather than the experience of individuals (transactions).

As used in IMSIM, transactions represent the simulation program workload. Thus, transactions are injected into the model to represent job requests, whether generated by a random function or read from an input script. These job transactions then cause other transactions to be produced to represent the steps (tasks) of the job, which in turn cause other transactions to be produced to represent the routines, data blocks, and messages which comprise the job. The most important transactions are those which represent tasks and messages. The task is the primary unit of work for a processor, while the message is the unit of work for a data link. These transactions require other simulated system resources in order to occupy processors or data links. Thus, a processor is only employed on a task (i.e., is acquired by a transaction) when certain routines and data blocks have been placed in memory, while a data link is acquired for message transmission only when the source and sink for the message are also accessible.

IMSIM incorporates a deterministic association between message transmissions and task processing; i.e., on the assumption that the purpose of a task is to transform input data to output data, each message transmission implies an amount of task processing. Similarly, an amount of processing implies completion of either the analysis of an input message or the

* In IMSIM, transactions are used to represent the job flow as the model proceeds from task to task, and the transmission of information over data lines.

preparation of an output message. Thus, whenever messages are delayed, processing may be delayed, and when processing is delayed, message transmission may be delayed. (It should be noted here that the model design prevents a stalemate in which both messages and tasks are waiting for each other).

Transactions also represent the job-steps and task types described by the model user. These two inputs are not really distinct classes, but rather a separation of task characteristics into two convenient groups. Job-step inputs specify the interrelationships of tasks within each job, while task inputs specify the tasks in more detail, including the elements and messages involved in performing the tasks. Transactions representing job step and task types are stored as "prototypes" which may be copied as often as required to develop a job in response to a job request.

Still other transactions are created during the development of a task environment to represent the routines, data blocks, and messages required for a task. These transactions then move through the logical paths of the model, determining additional task characteristics (e.g., memories to be employed, processing time), and at times take on the character of an executive task to represent system overhead functions in connection with task initiation.

The concept of IMSIM is a computer system, centered around the memory units which will provide the workspace for data to be processed and instruction for controlling equipment. The equipment consists, on one hand, of up to 20 computer processors which execute the vast majority of stored instructions and operate on the data contained in the memories, and, on the other hand, the data transmission links and peripheral units which send, receive, and store data. To provide for maximum flexibility in configuring the system components and to allow for representation of special structures such as multiprocessor and federated computer systems, the model permits each memory to be connected to any or all processors and to as many as 28 data links, and for each peripheral unit to be connected to as many as 28 data links.

The entire simulated system can be subdivided into as many as six subsystems, or "virtual machines". A virtual machine is characterized by a set of memory units and processors, configured so that every memory unit is addressable by any processor of the virtual machine and is connected to the same set of data links as every other memory. The executive tasks are automatically replicated for each of the virtual machines. This means that whenever a task or interrupt requires executive service, an appropriate processor from the pertinent virtual machine is selected and applied to an executive task. This concept enables the model user to simulate a variety of computer configurations within the context of one overall generic system. Configurations such as federated, shared-file, direct-couple, central-peripheral, etc., can be developed, each with multiprocessor capability. The only functional restriction placed on these configurations is that all autonomous computers employ the same type of executive, i.e., they are all governed by the same executive algorithms.

All data transmission in the DPA is simulated in IMSIM using the concept of "messages". A message is defined to represent a series of transmissions between two components of the system over communication lines or channels which are represented by "data links". The data links are not explicitly specified in connection with messages; instead, they are determined dynamically by a user-specified executive algorithm. In essence, this executive algorithm is a search for paths between the source and sink associated with a message.

In the simplest case, the source and sink represent the system components which transmit and receive a message. For example, a command (or series of commands) could be typed in on a keyboard for direct entry to a data buffer within a computer memory. This process would be represented by a message with the keyboard specified as the source "device" and the buffer as a sink "data block". Using message input parameters, the model user can specify the number of transmissions (commands in the example) which are represented by the message, the source and sink, the length of transmissions, the interarrival time between transmissions, and the amount of computation involved in processing each transmission. The length, interarrival time, and computation time can be specified as functions of a random variable, if desired.

Two other characteristics indicate the relationship between a message and tasks which may refer to it: 1) The start time for the series of transmissions may be specified relative to the start of a job or to a task. 2) The nature of a task is defined as the dependence of message transmission on task execution; transmission may be completely independent, or it may be dependent on a particular task execution. In the first case, denoted as "source-driven", transmission is controlled strictly by an interarrival time function, and may result in a data loss if resources are not available. This type of message may be shared by several tasks, as in the case of telemetry data which may be recorded, sampled, and reduced simultaneously. In the second case, denoted as "sink-driven", interarrival times may be extended beyond the amount specified by the interarrival time function due to delays in acquiring input/output units, data links, or delays in task processing (representing either the preparation of an output message or analysis of an input message).

All message transmission is assumed to be under control of an executive I/O service function. The particular executive is determined by the virtual machine to which the task is assigned. Every transmission is initiated by acquiring a suitable processor for the I/O request-service executive task.

The workload, equipment and executive algorithm specification types utilized by the model user to construct models are summarized as follows:

Work Simulation:

- Type 1 - Jobs
- Type 2 - Tasks
- Type 3 - Routines
- Type 4 - Data Blocks
- Type 5 - Messages

System Specifications:

- Type 6 - Devices
- Type 7 - Memory Units ("local" memories)
- Type 8 - Storage Units ("auxiliary" memories; i.e., operating and mass memories)
- Type 9 - Processors
- Type 10 - Data Links
- Type 11 - Data Sets
- Type 12 - Configuration Hookup
- Type 13 - Executive Algorithms

5.4.2 DPA Representation

5.4.2.1 Applications Workload

As stated earlier, a complete simulation of the entire DPA configuration, including the simulation of all workload inputs (i.e., all processing loads imposed by all slots of the postulated commutation cycle) is not a practical approach. Such a procedure would usurp an inordinate amount of computer time, and would therefore not be an efficient use of available resources. For this reason, a representative sample of the projected CP workload was selected for simulation, under the assumption that if this sample compared favorably to expectations (that is, the numerical analyses), reasonable extrapolations of the simulation results could be made.

The sample case so selected consists of the first four slots of Slot Group 1, as shown in Table 5-1 (slot indents A, F, B, and G). Thus, the generators of the workload will be RACU 35, RACU 64, RACU 63, and RPU 15. Additional work per slot for the arithmetic unit is generated by the need for interpretation of workload messages and having the arithmetic processor perform the CP background tasks on a slot by slot basis. In the I/O processor, additional work per slot is engendered by two periodic tasks, communications background and data bus scheduling.

It should be noted that the computing load triggered by these four sample time slots is the same for the revised Phase B DPA configuration as it is for this postulated ADT DPA configuration (described in Section 2). Thus, the simulation approaches and the numerical analyses for these slots are representative of both configurations. It should also be noted that RACU 35 includes several subfunctions that require a 50 ms response time. This is the highest response time requirement for the MSS. Thus, simulation of the highest MSS sampling rates were performed in this effort.

The job flow through the system is thus represented by a series of tasks. For the arithmetic unit, scheduling of task performance is done by buffering and/or by predecessor-successor relationships indicated on the job inputs. In the I/O processor, scheduling of task performance is done by buffering or by time control using the simulated clock. Associated with each of these tasks are messages. The model is set up to equate one word as being one character in regards to transmissions and one equivalent add as being one time unit.

At this start of each simulation run, it was assumed that no backlog of activities exist. This assumption causes the arithmetic unit to sit idle until the first inputs are received from the I/O processor. When the I/O processor so triggers information to be transferred for the arithmetic unit, the AU then proceeds to process the data, and eventually the AU will generate additional transactions that signal task completion. This does not imply, however, that the AU and I/O units are simply waiting for each other's triggers to perform tasks on a sequential basis. Each processor attempts to perform as many tasks on a "simultaneous" basis as possible; i.e., if resources are temporarily in use, or if message triggers are in process, each unit will attempt to execute other tasks to optimize the efficient use of these processors.

5.4.2.2 System Specifications

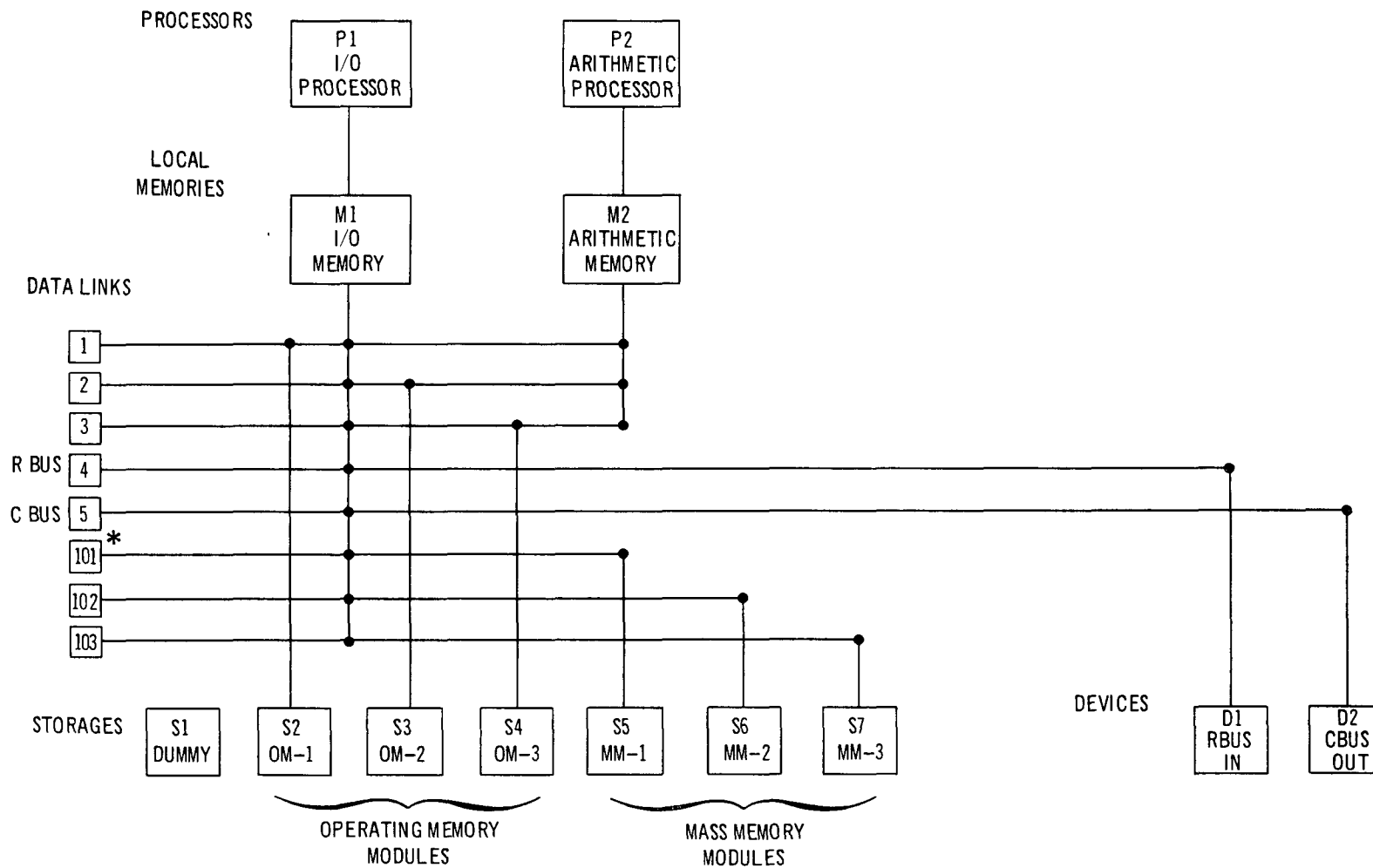
The model of the DPA is set up to represent a processing configuration where the arithmetic and I/O processing units are treated as two virtual machines. The interchange of information between the processing units is effected by means of buffers (IMSIM Data Sets) contained in the Operating Memory modules. To achieve this buffer interlinkage, the Operating Memories modules are treated as IMSIM storage devices with the characteristics of hi-speed memories; i.e., low access times, high transmission rates and random access. The Mass Memory modules are also treated as IMSIM storage devices, which have been given the characteristics associated with auxiliary memories. Only the local memories associated directly with the arithmetic and I/O processing units are explicitly handled as IMSIM memory units.

The channels connecting the I/O and arithmetic units to each of the three operating memories are assumed to be selector type channels, with a single channel for each of the Operating Memory modules serving both processing units. In this way only one of the processing units can utilize a channel at one time. The channels connecting the I/O processing units with the three Mass Memory modules are considered to be multiplexed, with burst mode capability. The performance characteristics of the I/O and arithmetic processing units, storages, data sets, data link channels and devices are contained on IMSIM forms. The interconnections among the above items are contained on an IMSIM input form and are illustrated in Figure 5-2.

Note that this configuration simulates essentially one-half of the CP; i.e., one AU, one I/O processor, and one set of associated equipments. Since both sides of a CP will essentially perform the same functions, it is not necessary to simulate the entire CP to verify the adequacy of the DPA.

5.4.2.3 Executive Representation

Two distinct executives are assumed, one for the arithmetic unit and one for the I/O unit. The complete I/O executive is assumed to be in residence in the local memory of the I/O processing unit. For the arithmetic unit, it is assumed that a nucleus executive is always in residence in the arithmetic unit's local memory and that additional executive capability would be obtained from Operating Memory storage when required. In terms of task scheduling, it



* Data links 101-103 are subchannels of a multiplexed channel

Figure 5-2. IMSIM Implementation of CP Configuration

is assumed that the arithmetic unit executive performs its assignments on a serial basis; that is, it performs all input or output processing for one processing load before it takes on the next load.

For both executives, it has been assumed that performance of both "pre and post" interrupt processing requires ten instructions each. In addition, fifteen instructions have been assumed as being required for both the arithmetic and I/O unit executives for switching between processing tasks. These assumptions were employed in the generation of associated tasks and messages for these processes.

A summary of the executive options is as follows:

(1) Algorithm 1 - Transmission Path Selection

Choose the first suitable link, whether in use or not, and wait till it is available (when necessary).

(2) Algorithm 2 - Virtual Memory Allocation

No virtual memory space consolidation will be performed.

(3) Algorithm 3 - Task Scheduling

Task scheduling will be performed in accordance with assigned priorities.

(4) Page Swapping

No page swapping will be performed (all required routines and memory data blocks are established in virtual memory prior to commencement of the slot workload transactions).

(5) Not applicable (devices, storage units, and virtual machines are selected for tasks by discrete specifications within other input forms).

In addition, the I/O processor is capable of responding to I/O and service request interrupts, while the AU responds to I/O, service request, and bounds fault interrupts. These executive functions are employed by IMSIM in the execution of simulation runs, and are reflected in utilization statistics.

5.5 SIMULATION RESULTS

As discussed earlier, and as summarized in Table 5-2, it is apparent that more than 4 time slots (16 ms) would be required for the arithmetic processor to perform all of its required functions for those 4 slots. Operating at 750K operations per second, this processor could only execute 12,000 operations in 16 ms, whereas Table 5-1 indicates that 12,206 operations would be required to process all inputs generated by RACU's 35, 64, 63, and RPU 15 during this time period. Thus, the simulation has been designed to

investigate the amount of time required to process this temporary saturation condition, so as to lead to conclusions concerning the adequacy of the DPA. (As Table 5-2 and the accompanying text illustrates, the total estimated backlog can easily be handled by this processor in a one-second cycle, with about 20% reserve capacity). Therefore, this simulation has been designed to investigate the amount of time required to process this temporary saturation condition, so as to lead to conclusions concerning the adequacy of the DPA. (As Table 5-2 and the accompanying text illustrates, the total estimated backlog can easily be handled by this processor in a one-second cycle, with about 20% reserve capacity). Therefore, this simulation is designed to investigate the nature of the I/O processor/arithmetic unit interaction and the capability of the arithmetic unit to accommodate temporary overloads. It should be repeated, however, that the overload projected here could be alleviated by reallocating device/slot assignments, rather than placing heavy requirements on the first slot group. However, since this worst-case approach provided an excellent test case for the simulated DPA configuration, the "saturation approach" was retained for simulation purpose.

As shown in Table 5-2, 9 time slots, or 36 ms, were used to complete a simulation run*. (As will be seen shortly, this time was mainly due to continued periodic operations of the I/O processor, rather than operations in the arithmetic unit. The AU actually completed its duties in 5 time slots). Pertinent statistics regarding the execution of the model during this period are as follows:

(1) Task Execution

MAXIMUM NUMBER OF TASKS IN PROGRESS SIMULTANEOUSLY	15
AVERAGE NUMBER OF TASKS IN PROGRESS SIMULTANEOUSLY	0.22

Of the total number of tasks simulated (29), over half (15) were in process at one time. Thus, between the AU and I/O, at least one point was reached where over 50% of the tasks were awaiting resources, awaiting the completion of other tasks, or were in execution. However, on the average, only 0.22 were in simultaneous progress during the 9 slot period. This is not surprising, since for 4 time slots, no tasks were performed on the AU and only one of the I/O; therefore the resultant long term average can validly be expected to be much less than 1.0 for this 36 ms run.

* In actuality, numerous simulation runs were performed, with modified workloads. The results presented here summarize the output of the most illustrative of these runs. (Preliminary runs were first made to check out the hardware configuration by executing "serial" processor runs; that is, the I/O processor would trigger a task for the AU, then remain inactive until the AU completed the task and signalled the I/O processor for more work. These initial runs served to verify the adequacy of certain hardware aspects, but they did not test the interactive effects of both processors operating on simultaneous tasks. Thus, more sophisticated workloads were progressively employed to investigate parallel processing, as well as to construct and verify special output reports to tabulate DPA results).

Table 5-2. Workload Summary

DPA WORKLOAD SIMULATED	4 time slots (16 ms)
ANTICIPATED (ARITHMETIC PROCESSOR @ 750 KOPS) LENGTH OF TIME TO PROCESS THESE 4 SLOTS.	> 4 time slots (12,206 operations)
LENGTH OF SIMULATED RUN	9 time slots (36 ms)

EXECUTIVE TASKS SIMULATED (Internal to IMSIM)	5
TASKS FOR SIMULATING VIRTUAL MEMORY LOAD	2
SIMULATED APPLICATIONS TASKS	22
TOTAL TASKS	29

ARITHMETIC PROCESSOR PROTOTYPE ROUTINES	2
I/O PROCESSOR PROTOTYPE ROUTINES	6
TOTAL PROTOTYPE ROUTINES	8

ARITHMETIC LOCAL MEMORY DATA BLOCKS & BUFFERS (SHARABLE)	2 blocks (3000 total words)
I/O LOCAL MEMORY DATA BLOCKS & BUFFERS (2 SHARABLE, 4 DEDICATED)	6 blocks (3500 total words)
TOTAL LOCAL MEMORY DATA BLOCKS & BUFFERS	8 blocks (6500 total words)

MESSAGE PROTOTYPES	48
TOTAL MESSAGE TRANSMISSIONS	55

(2) Operating Memory Utilization

<u>OPERATING MEMORY MODULE 1 (I/O→A COMM BUFFER STORAGE AREA)</u>	
NUMBER OF TRANSMISSIONS	17
AVERAGE TRANSMISSION TIME	0.10 ms
MAXIMUM DATA CONTENT	40 words
AVERAGE DATA CONTENT	0.05 words
<u>OPERATING MEMORY MODULE 2 (A→I/O COMM BUFFER STORAGE AREA)</u>	
NUMBER OF TRANSMISSIONS	17
AVERAGE TRANSMISSION TIME	0.03 ms
MAXIMUM DATA CONTENT	15 words
AVERAGE DATA CONTENT	0.19 words
<u>OPERATING MEMORY MODULE 3 (BUFFER FOR RECEIVING TRANSFERS FROM M.M.)</u>	
NUMBER OF TRANSMISSIONS	2
AVERAGE TRANSMISSION TIME	0.28 ms
MAXIMUM DATA CONTENT	1000 words
AVERAGE DATA CONTENT	992.45 words

These statistics detail some of the characteristics associated with the three modules of operating memory. Again, "averages" are computed over the full 36 ms run time. Thus, the low averages for OM modules 1 and 2 indicate that these buffers were occupied for a very small percentage of the run. However, the OM 3 buffer was full for virtually all of the 36 ms. Further inspection of message transmission statistics showed that this indeed was the case: the buffer was filled early in the simulation cycle, and remained so until associated periodic message transmissions were terminated at the end of the run.

(3) Mass Memory Utilization

<u>MASS MEMORY MODULE 1 (ROUTINE STORAGE AREA)</u>	
NUMBER OF TRANSMISSIONS	2
AVERAGE TRANSMISSION TIME	1.82 ms
<u>MASS MEMORY MODULE 2 (COMMUNICATIONS AREA)</u>	
NUMBER OF TRANSMISSIONS	3
AVERAGE TRANSMISSION TIME	1.90 ms

The two pertinent modules of mass memory exhibited the statistics shown above. As with OM modules, the average transmission times exhibit the lengths of time to transfer required amounts of data to appropriate elements of the CP at specified transfer rates.

(4) Data Link Utilization

<u>DATA LINK 1 - OM-1 TO AU, I/O</u>	
NUMBER OF TRANSMISSIONS	22
AVERAGE TRANSMISSION TIME	0.37 ms
<u>DATA LINK 2 - OM-2 TO AU, I/O</u>	
NUMBER OF TRANSMISSIONS	17
AVERAGE TRANSMISSION TIME	0.03 ms
<u>DATA LINK 3 - OM-3 TO AU, I/O</u>	
NUMBER OF TRANSMISSIONS	2
AVERAGE TRANSMISSION TIME	0.28 ms
<u>DATA LINK 4 - R-BUS</u>	
NUMBER OF TRANSMISSIONS	8
AVERAGE TRANSMISSION TIME	0.13 ms
<u>DATA LINK 5 - C-BUS</u>	
NUMBER OF TRANSMISSIONS	2
AVERAGE TRANSMISSION TIME	0.02 ms
<u>DATA LINKS 101-103 - MM-1,2,3 TO I/O</u>	
NUMBER OF TRANSMISSIONS	4
AVERAGE TRANSMISSION TIME	5.64 ms

As above, average transmission times reflect the times required to transfer appropriate amounts of data at specified transmission rates. Note that the relatively slow multiplexed channel connecting mass memory modules to the I/O processor require comparatively long average transmission times. This in part reflects the larger amounts of data that are required in MM→I/O transfers.

(5) Processor Utilization

<u>ARITHMETIC PROCESSOR</u>	
TIMES USED	117
TOTAL UTILIZATION	16.23 ms
<u>I/O PROCESSOR</u>	
TIMES USED	99
TOTAL UTILIZATION	1.47 ms

The above summary itemizes the overall utilization of the two processors for the 36 ms period, each operating at 750K ops per second. Note that the AU total utilization is within the expected range of 16-20 ms. Note also, that although the I/O processor is employed almost as often as the AU, its total utilization is considerably lower than the AU. This is partially because the I/O does not perform the relatively long applications processing tasks of the AU, but generally serves to transfer and format data within the CP. The low utilization of the I/O processor also suggests that a speed of 750 K ops per second is quite high for this unit, and that a lower speed or a reallocation of tasks among these two processors is desirable.

As an independent check on arithmetic processor statistics, special IMSIM output reports were designed to generate slot-by-slot utilization statistics for the run. A summary of these outputs is tabulated below and is illustrated in Figure 5-3.

ARITHMETIC PROCESSOR SLOT STATISTICS

SLOT	MAXIMUM COMPUTED LOAD *	ACTUAL OPERATIONS	PER CENT UTILIZATION **
1	3012	1965	65.5%
2	7435	2415	80.5%
3	769	3000	100.0%
4	990	2811	93.7%
5	—	1917	63.8%
7-9	—	—	—

I/O PROCESSOR SLOT STATISTICS

SLOT	MAXIMUM COMPUTED LOAD	ACTUAL OPERATIONS	PER CENT UTILIZATION **
1	NOT CALCULATED	240	8.0%
2		356	11.9%
3		140	4.7%
4		144	4.8%
5		30	1.0%
6		30	1.0%
7		20	0.67%
8		80	2.7%
9		25	0.83%

*See Table 5-1

**Percent Utilization = $\frac{\text{Actual Ops}}{3000 \text{ Possible Ops}}$

5-24

SD 72-SA-0114-4

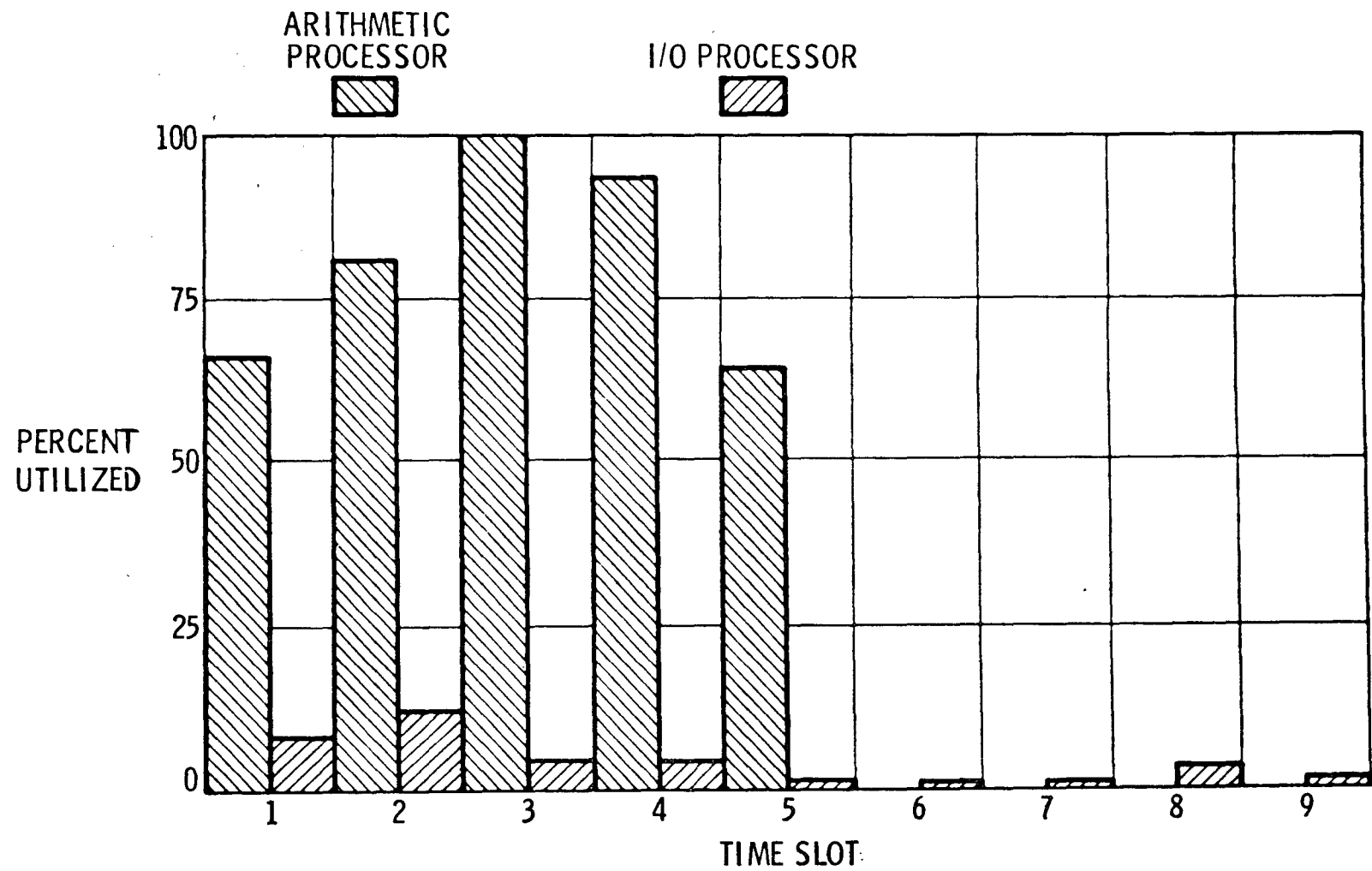


Figure 5-3. Processor Utilization

6.0 DPA REDUNDANCY STUDY

6.1 INTRODUCTION

The application of redundancy to the DPA stems from the failure criteria established for the MSS (Modular Space Station). The basic guidelines and constraints used in establishing these criteria were crew safety and mission continuance. The criteria are applicable to all MSS functions and as such dictate the following levels of functional classification:

1. non-critical functions
2. non-time critical functions
3. time critical functions

For each level of classification the following fault tolerance capabilities are required:

1. non-critical functions must fail safe following the first failure
2. critical functions must
 - a) be operational following the first failure (fail operational);
 - b) provide reduced performance subsequent to a second failure (fail degraded);
 - c) provide crew survival for 96 hours subsequent to a third failure (fail emergency).

The difference between time critical and non-time critical functions is the response time. Time critical functions require active (on-line) redundancy while non-time critical functions may be satisfied with standby redundancy (i.e., a functional replacement within some time period). This task was directed toward applying these criteria to the DPA concepts and recommending a satisfactory operational system.

A DPA redundancy configuration and an operational concept is recommended. The recommendation utilizes a multiprocessor, multi-computer organization with an interconnecting 4 channel data bus system. Rationale and tradeoffs are presented in support of this recommendation.

6.2 DPA REDUNDANCY CONFIGURATION

The recommendations given in this section are based solely on meeting the fault tolerance criteria specified for each level of functional criticality.

As stated previously, all MSS functions are assigned one of three levels of criticality, each having the following criteria:

<u>Level</u>		<u>Criteria</u>
(1) Non Critical	:	Fail Safe
(2) Non Time Critical	:	Fail Op, Fail Degrade, Fail Emergency
(3) Time Critical	:	Fail Op, Fail Degrade, Fail Emergency

By its nature the DPA executes many functions of all three levels of criticality and is therefore constrained to all of the above criteria. The DPA is tailored to take advantage of those criteria of lesser criticality. To achieve this, the computational (processing) requirements are split between two processors, each of which is organized to operate as either a multi-processor or a multi-computer. As such both processors will be located in separate isolatable volumes and, in general, both processors will have the capability of backing the other one up. In the case of time critical functions on-line backup (active redundancy) is provided, while off-line backup (standby redundancy) is provided for the non-time critical functions. In the case of non-critical functions the processors will fail safe noting that the actual action and reaction of the DPA relative to any one particular subsystem is out of scope for this report. That is, relative to any one particular subsystem insufficient information is known at this time as to whether the DPA should,

1. discontinue performing the function and notify operator;
2. execute a power-off command or standby sequence and notify operator;
3. execute a set of "pre-canned" tests for fault isolation;
4. switch in or request a possible backup unit;
5. take some other action.

In any case all of these functions and more can be performed, the details of which are recommended for some future study. The purpose here being to recommend a DPA redundancy configuration for meeting the failure criteria and to describe the operational sequence in the event failures are detected within the DPA. In addition, general recommendations for interfacing subsystems relative to their classifications are provided.

The redundancy configuration recommended herein is presented in Figure 6-1. This concept differs somewhat with the preliminary DPA configuration and the digital data bus configuration (see Volume III). The proposed recommendation requires all 4 channels of the data bus to be accessible from each of the various stations modules.

CP-1 (Typical For CP-2)

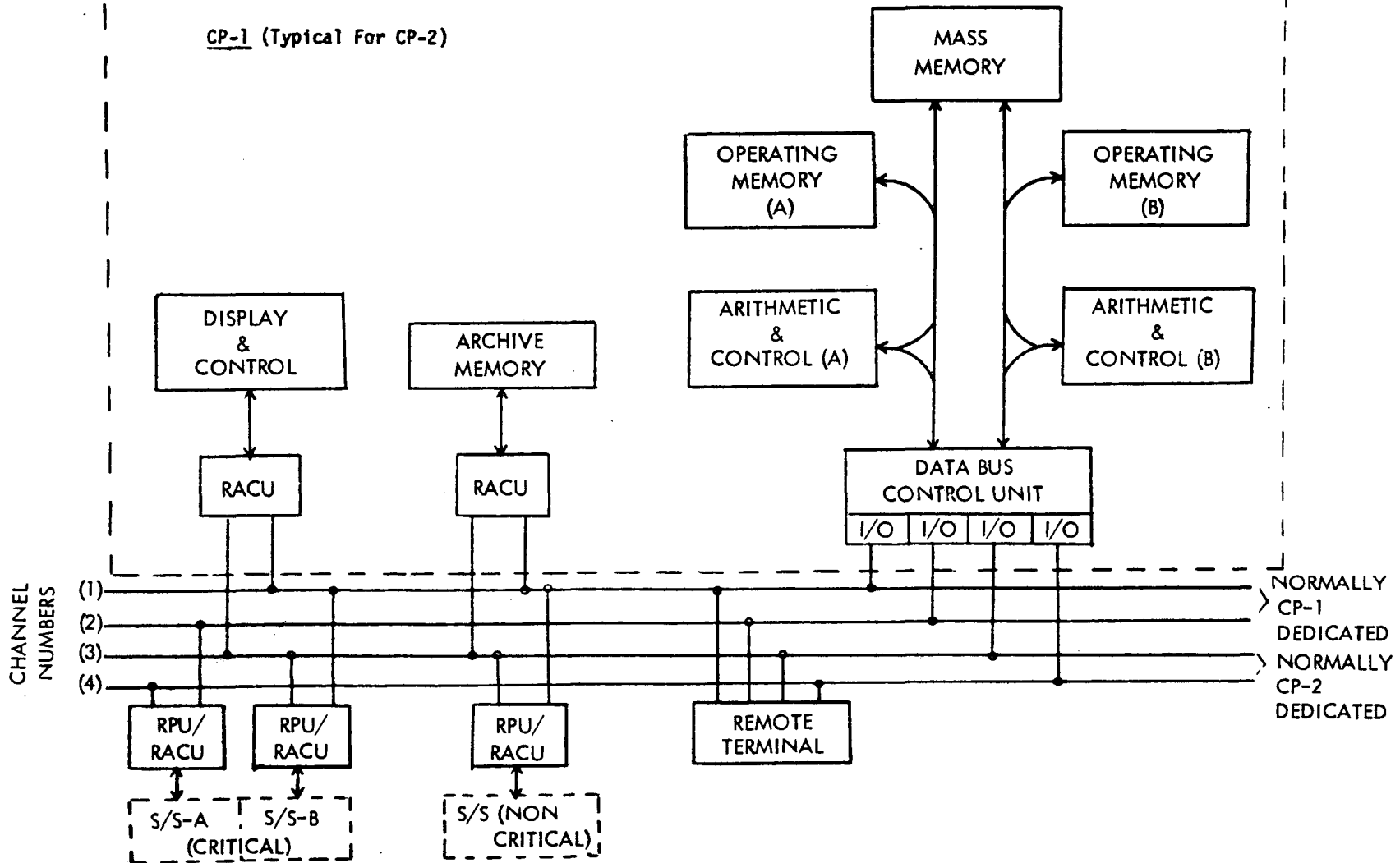


Figure 6-1. Basic Recommended Redundancy Configuration

The configuration, as shown, consists of two centrally located processors, one in each isolatable volume. The two processors are interconnected with each other and interfaced with the various subsystems through a 4 channel data bus system. When operating in the primary mode (no faults detected) each processor is in control of two of the four channels, each channel being independent of the others. As shown, each processor is provided a data bus control unit for this purpose. The amount of added redundancy applied to each processor is consistent with only the amount needed to provide the means for performing a comparative analysis for on-line critical functions and for reconfiguration. This can be accomplished with the two operating memories, two arithmetic units, plus the four I/O's as shown with both processors. The constraint is that all on-line instructions for critical functions are permanently stored in the operating memory. If this is not feasible, then an additional Mass memory will be required for each processor.

There is one further redundancy illustrated in Figure 6-1, the remote terminals. Since both central control and display actions are simplex in nature, the addition of at least two remote terminals provides the necessary backup plus the added capability of distributing DPA monitoring and control to other areas within the spacecraft.

Before going into the operational concept (i.e., response to faults), the problem of error detection is discussed and recommendations for detecting errors is presented.

6.2.1 Error Detection

On the surface, the application of redundancy the DPA appears to be for reconfiguration purposes only; however, this same redundancy applies to meeting the design goals for error detection necessary for fault tolerance systems.

In many large complex systems the amount (cost) of error detection is directly proportional to the amount of return as related to efficiency and down-time. Efficiency being related to the measure of time between the occurrence of an error and its detection while down-time is the measure of turn-around time for maintenance and repair. If a typical system, for example the DPA, is constrained to be fault tolerant (as in this case) and furthermore, must meet a maintenance requirement consistent with In-Flight Replaceable Units (IFRU's), then a completely different design goal must be used for the implementation of error detection. That is, a design goal approaching 100 percent error detection is desired plus having the added capability of isolating to no more than 3 IFRU's with a maintenance and logistics concept (man-in-loop) consistent with selecting which one of the three failed. This requirement will result from the relationship between MTBF and MTTR versus the number of IFRU's to be handled.

The attempt, here, is to establish a baseline concept for an error detection system, keeping in mind the desired design goal. However, it is very unlikely that this will be achieved while being cost effective. Furthermore, to perform a complete error detection analysis for the DPA and evaluate which of the many techniques is most optimum is out of scope for this report. Even so, an error detection system is the first building block

necessary in establishing a baseline operational concept. Thus, in the following paragraphs, the errors to guard against are identified, a means for detecting the errors is given, and the required response is provided.

6.2.1.1 Identification of Errors

The major errors to guard against, concerning the DPA, are identified as follows:

1. Operator/Program errors
2. Data transmission errors
3. Storage media errors
4. Equipment errors
 - a. Solid errors
 - b. Intermittent errors
 - c. Error detection errors
 - d. Power and cooling faults

For this system all programs loaded in the DPA and executed will be assumed error free. That is, all programs affecting MSS operations will have been checked and re-checked on the ground under "almost" identical conditions prior to being loaded in the actual DPA. Furthermore, any new programs added in-flight will be under direct scrutiny of the onboard personnel and supervisory program during the acceptance phase of such programs.

The remaining errors are self-explanatory noting that data transmission errors encompass all errors between the DPA and the subsystems, but not the subsystems themselves.

The errors identified thus far are all attributable to the DPA itself, That is, in the event any one of these errors are detected, any corrective action to be taken is made to the DPA. In essence, there exists another class of errors for which the DPA must react. Such errors would be attributed to the subsystems in the performance of functions and, any corrective reaction taken by the DPA would have to be in conformance with that subsystem's level of criticality, thereby its failure criteria and corrective action. Since this type of information has not yet been specified in sufficient detail such reactions will be left for future studies. If the DPA is interdependent with a subsystem in the performance of a critical function and an error in this loop occurs as a result of the DPA, detection and response would fall into the first set of errors identified.

6.2.1.2 Means for Detection

For the system in question the designer has at his disposal four basic means to the solution of detecting a given error:

1. Hardware
2. Software
3. Man-in-the-loop
4. Any combination of 1, 2, or 3

The problem is not so much in selecting an approach but once having selected an approach there exists a multitude of techniques for the detection of any one particular class of error. Thus, the designer must wade through the various techniques and select only those most applicable. Having selected the more applicable techniques for the various classes of error, an analysis must then be conducted in order to determine which combination of techniques is best suited to meeting the desired design goals while being cost effective. Obviously this amount of detail is out of scope for this study. Therefore, the method used here is to apply techniques in all of the approaches which have been successfully used in the past and to capitalize on the recommendations made in the study on "Data Acquisition and Control Redundancy Concept" (see Volume III). Furthermore, this method will attempt to provide sufficient overlap between the various approaches and techniques such that errors missed by one technique are detected by another.

6.2.1.2.1 Operator Errors. Operator errors are recommended to be detected and controlled by means of hardware, software, and hardware/software combinations.

Memory protection is to be implemented by hardware. That is, the operator(s) will be provided the capability of keying data into fixed memory locations only (hardwired address from location A to location B). All data entered will then be under software control and thereby checked by the routines and/or programs modules for which the data are intended. The supervisory program is required to be self protective against any changes that may affect crew safety and/or mission continuance. Transmission errors will be discussed in a subsequent paragraph.

The response of the DPA to operator errors will be to notify the operator and identify the error.

6.2.1.2.2 Data Transmission Errors - Data transmission errors are defined as those errors occurring in the transfer of data in the four channel data bus system. In the recommended configuration, two channels are dedicated to each processor when operating in the normal mode (no faults). For this case the following means for error detection is recommended.

1. Error checkers preceding drivers and following receivers as illustrated in Figure 6-2. These may be implemented as either Longitudinal and Vertical Redundancy checkers (sometimes called serial/parallel parity checkers) or as one of the class of Polynomial checkers. For the present the LRC and VRC will be used.

"Typical Between DPA & Subsystems"

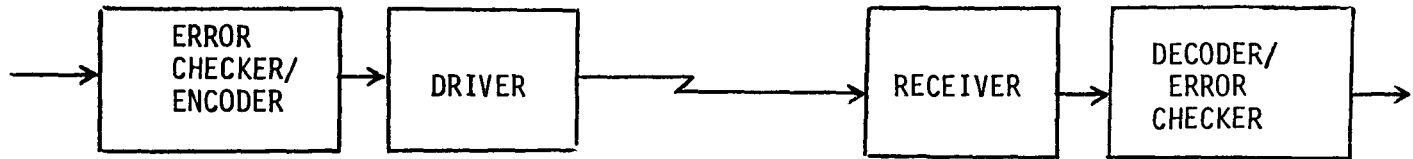


Figure 6-2. Data Transmission Error Checkers

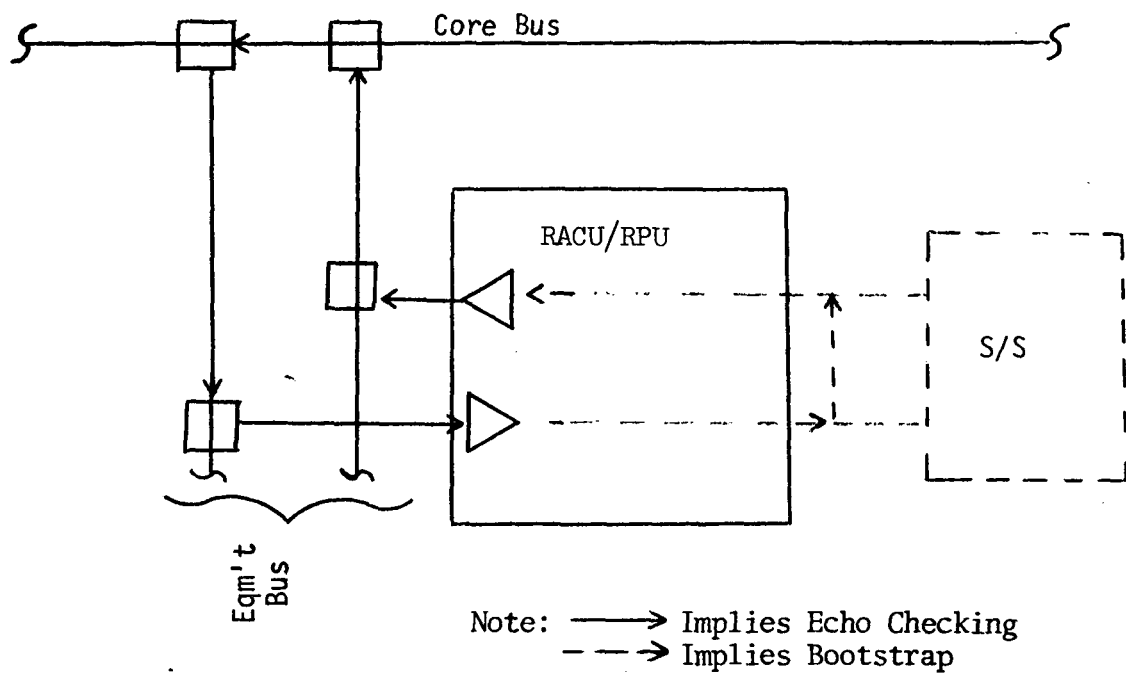


Figure 6-3. Short Message Echo Checker



2. Periodic "Short Message" echo check. This will be used as a subsystem to-core data bus self test illustrated in Figure 6-3. As such, this provides the added capability of independently error checking and isolating terminals without added hardware costs. It is assumed that CP-1 and CP-2 perform this check periodically while subsystems perform an echo check prior to executing the transmission of any message.
3. Simultaneous data transfers are recommended between the central processors and those subsystems performing critical functions. This is consistent with operating the central processors in the multi-computer mode. In this case the central processor provides the capability of comparative analysis in order to approach a 100 percent error detection design goal. However, there are two problems using this approach. The first is that of isolating to an IFRU in the event of a fault. This is a necessary requirement in order to minimize the mean-time-to-repair. The second problem is that no two subsystems interface with the DPA in exactly the same manner. For example, the RCS and IMU's are designed to detect faults and reconfigure within themselves. Thus, based on the assumption that critical subsystems perform a comparative analysis on the data received (which can only be determined when the details of the subsystem mechanization are known), the following capability is recommended. Capitalizing on the Echo check described above the "Short Message" is recommended to be boot-strapped through the RACU and/or RPU I/O with the subsystem prior to checking the data bus interface as illustrated in Figure 6-3. In this manner a complete thru-put checkout can be made with little or no imposition on the central processor and with a minimum of hardware. Note this last recommendation (i.e., boot-strapping) is applicable to all interfaces.

6.2.1.2.3 Storage Media Errors. It will be assumed that the storage media (archive memory will be interfaced with the central processor in the same manner as a subsystem (i.e., via the 4 channel data bus) and thereby treated in the same manner. The only added recommendation is that programs and/or data read from this source are under the direct control of a supervisory program and thereby checked for integrity. This can be done using several acceptable techniques (e.g., either the LRC and VRC or a polynomial check).

6.2.1.2.4 Equipment Errors. The means recommended for the detection of equipment errors includes software, hardware, and the combinations of both. This is not to exclude the man-in-the-loop who provides the overall backup and who, having encountered two failures, must select which of the two last computers to use in the event of a third failure (assuming the first two failures has not been repaired).



The first recommendation is that error checkers be limited to the level of an IFRU. This is consistent with the maintenance and repair concept, and to go below this level would only result in unnecessary costs. Furthermore, it is recommended that the error detection system be designed to provide sufficient overlap such that only single fault detectors need be used (e.g., a simple parity checker or equivalent). These recommendations are based on having the added capability of performing comparative analysis in the processor and, if required, in the subsystem.

There are two types of errors that the checker can detect (assume the checker is not failed): a solid error or an intermittent error. To detect the difference, a combination hardware-software technique is used. The method recommended is normally referred to as roll-back. If an error is detected an interrupt is created and a retry is made. If the retry is unsuccessful the error is considered solid, otherwise, intermittent. If the processor is operating in the multi-computer mode, this information must be transferred from one computer to the other and acted on accordingly, i.e., time phased prior to comparing outputs. If an error is detected in the I/O, the good data are transferred normally followed by a flag to both receivers. This can be accomplished relatively easily having multiple access to all RACU's and/or RPU's.

To check the error checkers, a "canned" routine may be used periodically for injecting an error into the system for this purpose.

Finally, the processors are recommended to be operated in the multi-computer mode and comparative analysis be performed on the data. This is needed whenever the added assurance of meeting the 100 percent error detection design goal is required.

It will be assumed that power and cooling are redundant and provisions for failure detection are implemented within these systems.

6.2.2 Operational Concept

The operational concept referred to herein is concerned only with the operation of the DPA 's response to detecting an error (fault) and correspondingly satisfying the fault tolerance criteria. The operational concept is consistent with the redundancy configuration given in Figure 6-1. For simplicity it will be assumed that initially one processor (CP-1) performs the MSS operations while the other processor (CP-2) performs experimental functions. In this way the operational concept can be described for CP-1 noting that the converse is applicable to CP-2.

6.2.2.1 Normal Operation

Normal Operation is defined as operating in a "no fault condition". Operating in this condition the DPA and subsystems will be performing non-critical and critical functions. In the performance of non-critical functions the DPA will be operating in the multiprocessor mode. In this mode, CP-1A and B (A & B refer to independent arithmetic units contained within each processor, see Figure 6-1) will be sharing a common memory bank in the performance of the required non-critical arithmetic and logical operations.

The transfer of data between CP-1 and its associated subsystems will be conducted over the two data bus channels dedicated to CP-1. In this manner non-critical functions can be executed efficiently in a minimum amount of time. In this mode, all of the error detection capability described, with the exception of comparative analysis by the processor, is applicable. In essence, the DPA is mechanized to satisfy only the criteria of failing safe when operating in this mode (i.e., not overly designed).

In the performance of critical functions, CP-1A will be operating independent of CP-1B. This is referred to as multi-computer mode. In this mode both arithmetic and control units will be performing identical functions simultaneously, each operating from independent memory banks with the resulting data compared for errors. If no errors are detected, both sets of data are transferred simultaneously on separate channels. This allows both receivers at the subsystem level, access to independent data; one receiver receives one set, the other receives the other set. This is in conformance with the recommendation that if comparative analysis is required at the subsystem level that it be performed as close to the point of criticality as possible. That is, performed within the subsystem where the criticality exists. Conversely, data transferred from critical subsystems will, in general, be transmitted as two independent sets over the two dedicated channels. The added features of this mode over the multiprocessor mode is the comparative analysis on the independent data which is necessary in trying to achieve the 100% error detection design goal.

For purpose of this concept it is assumed and recommended that the supervisory program be located in both CP-1A and B and that both operate independently from a common real time interrupt (external clock). As such that will allow synchronization of modes where the modes are time scheduled. That is, a delta (T_1) time for non-critical functions (multi-processor mode) and a delta (T_2) time for critical functions (multi-computer mode).

6.2.2.2 Single Fault Reconfiguration

The faults referred to here and in the following paragraphs are those faults which are cause for possible reconfiguration. Such faults include those classified as data transmission errors, equipment errors, and errors derived from comparative analysis.

In general, faults occur in four major areas: subsystem, RACU and/or RPU, data bus, or a central processor. Taking one area at a time, when operating in a multi-processor mode, and imposing an error the following responses are recommended:

- Subsystem: Notify operator and respond in accordance with subsystem requirements.
- RACU/RPU: Notify operator; respond to subsystem; isolate fault using results of "short message" echo check in combination with transmission check; notify operator and retry on request to determine if failure was intermittent.

Data Bus: Notify operator; execute retry to determine if failure was intermittent; if not, isolate failure using "short message" echo checks, transmission error checkers; interrupt CP-2 and secure one of its two data channels; notify operator and continue normal operations.

Central Processor: Notify operator; execute retry to determine if failure was intermittent or solid and notify operator accordingly; if solid and not in mass memory (1) interrupt CP-2 for a critical function take-over (2) terminate CP-1A or B whichever has fault and continue operating on non-critical functions; if solid and in mass memory interrupt CP-2 to take over non critical functions (if CP-1 archive memory could be used this would be preferred).

When operating in the multi-computer mode and a fault is detected the following responses hold:

Subsystem: Notify operator and reconfigure in accordance with subsystem requirements or notify operator that the subsystem has reconfigured within itself.

RACU/RPU: Notify operator; determine if failure is intermittent or solid; if solid reconfigure as per subsystem requirements or in some cases (monitoring) notify supervisory program and reconfigure software to handle only one source of independent data; isolate failure and notify operator.

Data Bus: (same as multiprocessor mode)

Central Processor: Notify operator; interrupt other arithmetic and control unit and execute retry; if intermittent, continue; otherwise interrupt CP-2 to take over critical functions and to aid in isolating the failure was not detected by one of the checking techniques, in which case isolation is effected immediately; otherwise, CP-2 will check for differences in the data and, if different, authority is relinquished back to CP-1 and the subsystem reconfigured; if they are the same, the data are operated on and compared with CP-1A and B's results for isolation; the failed one is terminated and the other is assigned the non critical functions, noting that CP-2 will be performing the critical functions.

6.2.2.3 Second Fault Reconfiguration

To continue second and third fault reconfigurations for the subsystems and their interface with the data bus, is meaningless without more specific information. That is, the reconfiguration concept applied to the central processors and data bus is just as applicable to the subsystems without further information.

It is assumed, therefore, with the exception of a mass memory failure that CP-2 is performing critical functions and either CP-1A or B is performing non-critical functions at this point. This is acceptable since a failure in mass memory need only fail safe. When operating in this configuration it is recommended that periodically all three arithmetic and control units perform an identical "Canned" operation for comparison of performance. This adds to the confidence level that the CP-1A or B system is operating and, thus allow CP-1A or B to be used for isolating between CP-2A or B in the event of failure.

It is further assumed for the present that the first failure response is applicable for second failures in the subsystems and interface equipments, therefore only failures in the data bus and processor are discussed.

Data Bus with First Failure in Data Bus: Notify operator and a) if second failure is in CP-1 data bus, interrupt CP-2 for takeover of non-critical functions; continue reading critical data in CP-1 on CP-2 channels for isolation purposes in the event of a CP-2 failure.

b) if second failure is in CP-2 data bus channel, interrupt CP-2 for take over of non-critical functions and the remaining dedicated CP-1 data bus channel; continue reading critical data in CP-1 for isolation.

Data Bus with First Failure in CP-1: Notify operator and a) if second failure is in CP-1 data bus, switch to second channel and continue.

b) If second failure is in CP-2 data bus, interrupt CP-1 and take over data bus not in

Second Failure in CP-1: Notify operator; interrupt CP-2 to take over non-critical functions; initialize program for possible third failure (man-in-loop isolation).

Failure in CP-2 with First Failure in CP-1: Notify operator; interrupt CP-1 to isolate failure, if not detected by error checker; continue operating with C -1 executing non-critical functions and CP-2A or B critical functions; continue comparative analyses on "canned" program between processors; periodically perform comparative analysis on all data accessible from two independent sources; increase man-to-machine communications; execute all reconfigurations under operator control.

CP-1 Failure with First Failure in Data Bus: Notify operator; interrupt "good" CP-1A or B unit for take over of non-critical functions.

CP-2 Failure with First Failure in Data Bus: Notify operator; interrupt CP-1 for take over of critical functions and one of the two CP-2 data channels and to isolate between CP-2A or B if undetected by checkers; critical functions will continue being monitored by CP-2A or B for isolation purposes.

All of the second fault reconfigurations described are based on detecting a solid fault. This assumes that prior to reconfiguring a retry will be executed to differentiate between intermittent and solid faults. One precaution must be noted here. If a subsystem is required to perform comparative analysis and two successive data bus failures occur such that one of the two interface equipments are disallowed access to independent data, then that subsystem must be alerted to conduct its function based on a single set of data (see Figure 6-1).

6.2.2.3 Third Fault Reconfigurations

To provide the details for all possible third fault reconfigurations is somewhat out of scope for this report. To be more explicit, there are 18 basic reconfigurations if the order of data bus failures is neglected. On the other hand, if they are accounted for there are a possible 108 reconfigurations which should be described. The 18 possible basic reconfigurations are illustrated in Figure 6-4 noting that if data bus failures were ordered this would expand to 108 possibilities. Therefore, only the impact of the man-in-the-loop will be discussed relative to third fault reconfigurations.

Ideally, it would be desirable to automatically reconfigure through any three consecutive failures. With the present concept, however, this cannot be achieved for all cases and therefore configuration must be supported by the operator. The role of the operator can best be illustrated by assigning the following priorities as a function of the sequence of failures:

Priority One: 3 consecutive arithmetic unit failures

*Priority Two: 3 consecutive data bus failures

*Priority Three: Any combination of 3 failures involving either channels
 1 and 3 or 2 and 4 of the data bus

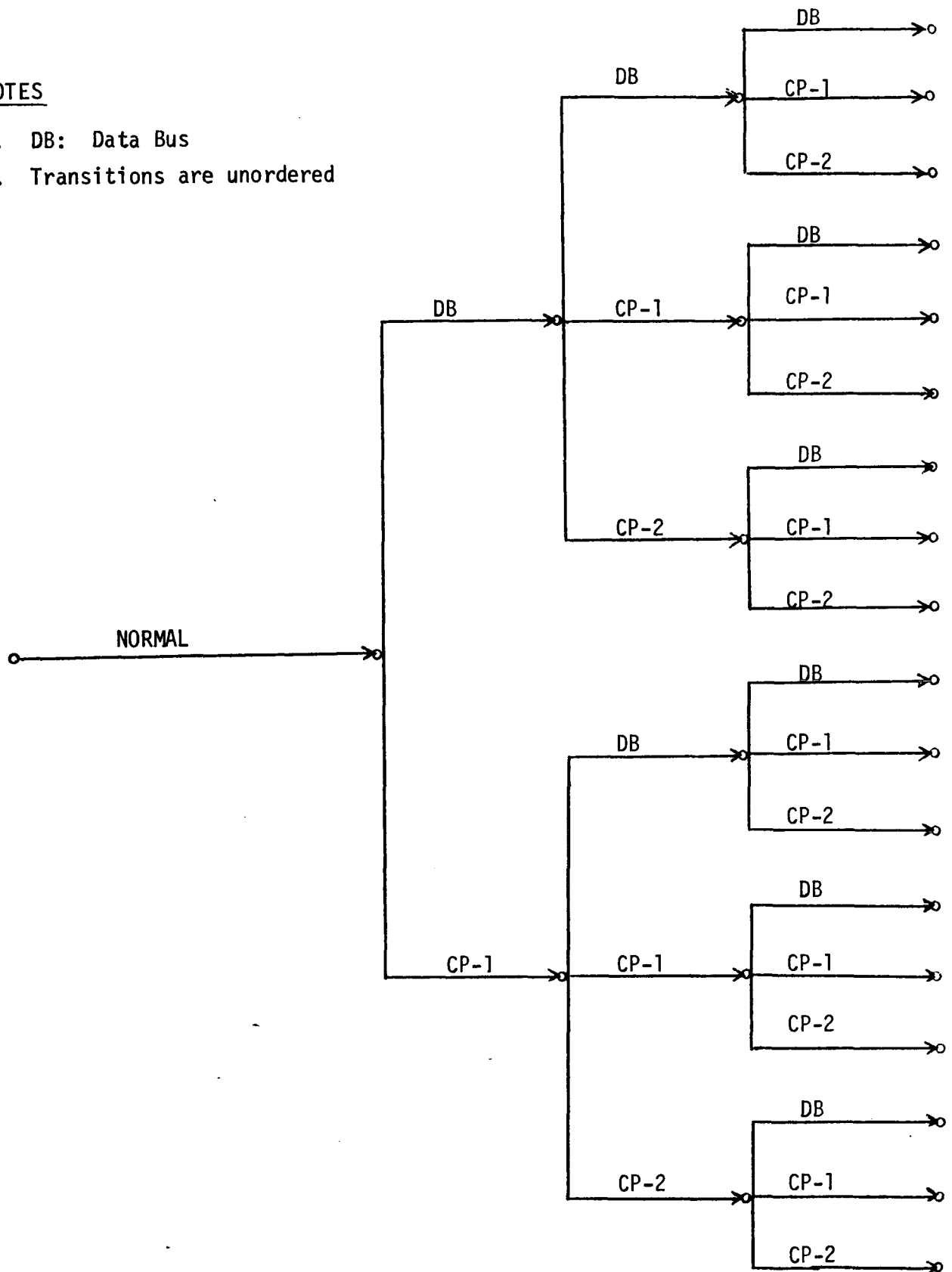
Priority Four: Any other combination of three failures

*Note that the "Two Fault Reconfiguration" case is designed to handle the event of two consecutive failures in channels 1 and 3 or 2 and 4 (i.e., 4 cases).

For priorities one, two, and three as listed, the operator is required to isolate the fault and initiate the reconfiguration for at least one of two possibilities in each case. His decision will be supported by the proposed error detection system, software design aids, and any added capabilities provided by the various subsystems. In the present concept this would involve 43 cases: 3 at priority one, 16 at two, and 24 at level three. For the remaining cases, which involves approximately 65, the DPA can automatically reconfigure and thereby reduce the burden on the operator. The responses required by the processor(s) would be typical of those described for the "one" and "two" fault reconfiguration cases. These responses can be elaborated on at the time when the subsystem reconfigurations are known in more detail and at that time the total system response can be described relative to

NOTES

1. DB: Data Bus
2. Transitions are unordered



NO FAULT

FIRST FAULT

SECOND FAULT

THIRD FAULT

Figure 6-4. State Diagram for Fault Transitions

detecting all errors. It should be noted that for many of those cases falling in priority 4, that a fourth failure could be tolerated with the man-in-the-loop concept.

6.2.3 Summary

The various errors to guard against the DPA were discussed. A baseline error detection system was developed based on the following criteria:

1. To detect errors consistent with fail safe criterion established for non-critical functions.
2. To provide a means for approaching a 100% error detection capability in the performance of critical functions.
3. To be compatible with the In-Flight Replaceable Unit concept for repair and maintenance.

The recommended DPA redundancy configuration was elaborated on. The two central processors (CP-1 and CP-2) were recommended to operated in both the multiprocessor and multicomputer modes. This being necessary in order to satisfy the following requirements while being efficient and cost effective.

1. Provide sufficient memory capacity and speed capability to perform the required MSS and on-board experimental operations simultaneously
2. Provide a design consistent with the failure criteria established for the MSS.

A preliminary operational concept was presented with the recommendation that the concept be expanded in future studies to include the details of the response to errors for the subsystems.



7.0 CENTRAL PROCESSOR STUDIES

7.1 CENTRAL PROCESSOR OPERATIONAL ANALYSES

This section presents the results of the DPA Central Processor Operations Analyses task. The objective of this study was to determine the impact of the MSS central processor operational use and software organization on the design of the hardware aspects of the central processor.

7.1.1 Software and Data Types

This study began with a review of the specific DPA computational requirements and general software production problems as they impact the architecture and hardware design of the central processor, with great emphasis on overall system cost effectiveness. Although very few computational characteristics, other than the processing of specific data structures, were found to bear directly on the choice of computer hardware, many general aspects of program execution and software production were found to be sensitive to hardware organization and design.

In attempting to extract basic processing characteristics that do have impact on machine design, it became apparent that what really mattered was the structure of the data elements involved in the computations, and the kind of arithmetic or logical manipulation to which they were subjected. The processing system can be designed to handle specific data types.

The scope of allowable data types is an important aspect in the software design. They represent the forms of information which are processed by the computer programs. The required data types depend to a large extent upon the intended use of the multi-processor system, and may impact the design of the software and hardware system.

The following data types have application to the space station:

1. Boolean. A Boolean data type is a variable which can assume only one of two values, true or false, on or off. An example of a Boolean data variable is an overflow bit in an arithmetic unit, or an execute bit in an I/O control word.
2. Bit strings. Bit strings are a collection of one or more binary bits. String data possess a length property. A bit string of length one may be considered a Boolean variable. However, a generalized bit string may be of any length. The entire string is an addressable entity. Bit strings are utilized to record status information, generate control information, and pass discrete information between software modules.



3. Scalars. Scalar variables are numbers represented in fixed or floating point formats. The exact format is not of importance at this time. Scalars, besides being used in arithmetic operations, are utilized to convey quantitative information such as O₂ pressure, fuel cell voltage, etc.
4. Vector. A vector is an array of scalars obeying the laws of vector algebra. It is represented by n-components within an n-dimensional coordinate system.
5. Matrix. A matrix is a rectangular array of M rows and N columns of MN scalar elements. A matrix may also be thought of as N vectors of dimensionality M. A matrix obeys the rules of matrix arithmetic. Matrices are utilized for various G&C functions, including coordinate transformation, and error coefficient matrices.
6. Character strings. A character is a non-numeric (in the sense of value) data type consisting of letters, numerals, or other symbols. Like a bit string, a character string consists of a variable number of characters addressable and manipulable as a single entity. A character string of length N consists of N individual character elements. The string "DISPLAY DATA" is a 12-character string. Blank is a legal character. Character strings are the main data type used for crew/computer interaction.
7. Pointers. This data type contains information about the location of another data type. Pointers are utilized as control mechanisms to develop generalization and flexibility in the software system. They allow the dynamic operation of the software system and provide a means for linking program modules, data modules and control modules. An example of the use of pointers is in a file directory where the file name is used as a key for retrieving the pointers which indicate the storage location of the file.
8. Name. This data type is the differentiating reference to similar data types. For instance, a matrix, scalar, bit strings are all referenced to by name. The name "STATE VECTOR 5" refers to a specific unique vector. No other vector in the system has the same name. However, it is possible to have a name equivalence, where the same data element possesses more than one name.
9. Array. An array is a collection of identical data types known by one name. All the elements within an array must possess some consistent attributes. For example, in an array of vectors all vectors must possess the same dimensionality. Every character string in an array of variable length character strings must possess the same maximum length.

10. Structures. A structure is an hierarchical organization of data which may contain other structures, arrays or individual data types. A structure need not consist of identical data elements. It may contain many levels. The outermost structure is called a major structure and is considered to be at level one. Minor structures are considered to be at lower levels, 2, 3, 4, etc. Each item in a structure possesses a name. If the name of a major structure is referenced, the entire structure including all subroutines and elements are addressed. If the name of minor structures is referenced, all the elements of the minor structure are addressed.

Structures appear in program organizations as well as file management situations. A major program containing sub-program modules which in turn reference other subroutines can be considered to be a structure. Another example of a major structure is a file. A file structure may contain many minor structures or subfiles which in turn may contain pages which can be considered to be arrays of addressable parts.

Table 7-1 indicates which data types are used for each operational software requirement. Besides the specific operational requirements 1 through 8, taken from Section 2.0, the overall software system includes other, nonoperational functions 9 through 12. The intent of the following paragraphs is to define these requirements in some detail.

1. Sequence and Control

The large number of operational programs which must be executed sequentially and on demand involves internal task scheduling, task queuing and priority control. Maximum utilization of the processor demands a multiprogramming environment which allows more than one program to be run in the same processor at the same time; e.g., if one job is waiting for an I/O request to be serviced a second job can be executed. Input/output and interrupt control are a part of this function.

2. Resource Allocation

The large amount of memory required for program and data and the high processing rates demanded by the space station have led to the consideration of a multiprocessor with an hierarchical memory organization containing four levels of memory: M_1 - local storage dedicated to a particular P; M_2 - operating memory accessible by all P and I/O modules; M_3 - the mass memory accessible via the I/O unit, and M_4 - the archival memory also available via the I/O unit and servicing both multiprocessors.

Table 7-1. Requirements and Data Types

Data Type*	1	2	3	4	5	6	7	8	9	10	Characteristics
Function											
1. G&C											
Exp. mod. update	X		X	X	X						(1) Highly mathematically oriented - extensive use of vector and matrix algebra
Shuttle alignment	X		X			X					(2) Wide dynamic range of numeric data
Terminal rendezvous	X	X	X	X	X	X					(3) Boolean variables and flags for logical decisions
Docking	X	X	X	X	X	X					(4) Real time control
2. EPS											
Solar array pointing control	X	X	X			X			X		(1) Control requires Boolean variables for logical decisions as well as bit strings and scalars to send control information.
Fuel cell control	X		X	X	X	X					(2) Scalars are used to monitor quantitative information.
Lighting control	X	X				X			X		(3) Arrays are employed to store large amounts of control information
											(4) Vectors and matrices are required for statistical analysis
3. ETC/LSS											
Pump and repress	X	X	X			X					(1) Control requires command words generated from bit strings and characters
CO ₂ management	X	X	X			X					(2) Monitoring requires scalars for quantitative information
Atmosphere control	X	X	X			X					(3) Boolean variables are required for logical decisions
Active thermal	X	X	X			X					(4) Food management involves arrays and structures as well as names and values (scalars)
H ₂ O management	X	X				X					
Food management	X	X	X			X		X	X	X	
Special LSS	X	X	X			X					
4. RCS											
H ₂ /O ₂	X	X	X			X					(1) Simple arithmetic
N ₂	X	X	X			X					(2) Decisions
Thrust valve function	X	X	X			X					(3) Monitoring and control
5. Crew											
R.T. medical data	X	X	X			X			X	X	(1) Monitoring, control and storage of data in arrays
N.R.T. medical data	X	X	X			X			X	X	(2) Utilization of structures for file management
Medical analysis			X	X	X	X			X		(3) Mathematical computations for analysis
6. Structures											
Docking	X	X	X			X					Monitoring and control functions
7. ISS											
Communications	X	X				X					Bit manipulation, character strings and logical operations

*See legend at end of table.

Table 7-1. Requirements and Data Types (Cont)

Data Type Function	1	2	3	4	5	6	7	8	9	10	Characteristics
7. <u>ISS</u> (Cont)											
Displays		X	X			X			X		Character and message generation Use of arrays
Mission management	X	X	X				X	X	X	X	Simple arithmetic operations file structures
8. <u>OBCO</u>	X	X	X	X		X		X	X	X	(1) Simple mathematical computations (2) Hardware interfacing via I/O (3) Bit manipulation for complex decision problems (4) Data comparison
9. <u>Sequences and Control</u>	X	X	X			X	X	X	X	X	(1) Logical decisions (2) Control of program structures (3) Queue control, use of pointers
10. <u>Resources Allocation</u>											
Memory control	X	X				X	X	X		X	(1) Paging, address control, directory searching
Processor control	X	X				X	X	X		X	(2) Priority control, real time program control (3) Very little arithmetic
11. <u>Memory Protection</u>	X	X	X					X	X		(1) Address comparison (2) Pointer control
12. <u>Failure Anticipation and Recovery</u>	X	X	X				X		X	X	(1) Data backup storage (2) Comparison verification (3) Pointer control (4) Complex prestored decisions for recovery

LEGEND (DATA TYPES):

- 1 Booleans
- 2 Bit Strings
- 3 Scalars
- 4 Vectors
- 5 Matrices
- 6 Character Strings
- 7 Pointers
- 8 Names
- 9 Arrays
- 10 Structures



The control of the transfer of information between these memory levels is a significant requirement. M1-M2 transfer will be under hardware or firmware control with a fixed paging type algorithm. M2-M3 and M3-M4 information transfer control can be accomplished in many ways through implementation of paging segmentation, or overlay techniques. Economy of memory, higher performance and flexibility introduced by incorporating a dynamic allocation and deletion scheme.

A second area of resources allocation is establishing the relationships between tasks and processors. It is possible to preassign tasks to processors. This assumes that complete information concerning task performance is known a-priori. The introduction of new tasks might require a new assignment strategy. A more general approach is to assign tasks to processors at execution time depending upon priority of the task and the busy status of the various processing elements, and is the approach usually considered in maximizing the capabilities of a multiprocessor organization.

3. Memory Protection

The large number of independent tasks requires that the tasks be allowed to access only those areas of memory to which they are assigned. This memory protection requirement helps reduce the propagation of errors. It may be implemented by a combination of software and hardware techniques.

Another area of memory protection which must be considered is the utilization of common data between many tasks. Tight control of this COMPOOL is required to provide temporary data lockout during write operations. Some data may only be read, other data may be modified by only a particular task, while a third category of data may be modified by all users. The illegal modification of data in the COMPOOL by unauthorized tasks must be detected and the task aborted.

4. Failure Anticipation and Recovery

The operational requirements allocate a significant amount of memory for OBCO functions associated with all the space station I/O equipment. These OBCO programs mainly deal with fault isolation and initial checkout.

Requirements exist for anticipating failures within the multiprocessor itself and recovering from failures after they occur. Sufficient backup storage must be made available so information is available to properly initialize redundant equipment and to take over in case of failure.

Associated with each entry in Table 7-1 are comments concerning the nature of each computation. Functions such as G&C or analysis programs are very mathematically oriented and use scalars, vectors, and matrices. Functions which

perform mostly monitoring and control functions require scalars for quantitative information, and generate control words by manipulating and concatenating bits and characters. Very complex decision processes require structures. Storage of large quantities of data can exploit the properties of arrays. Data types such as names and pointers are most useful for program and memory control functions.

7.1.2 Approaches to Fault Tolerance in the Multiprocessor

The choice of an effective fault tolerant design requires an investigation of error detection mechanisms, fault isolation logic, and recovery philosophy. These items will be discussed in general and their application to the operating units of the multiprocessor (P, M2, I/O, bus) will then be presented.

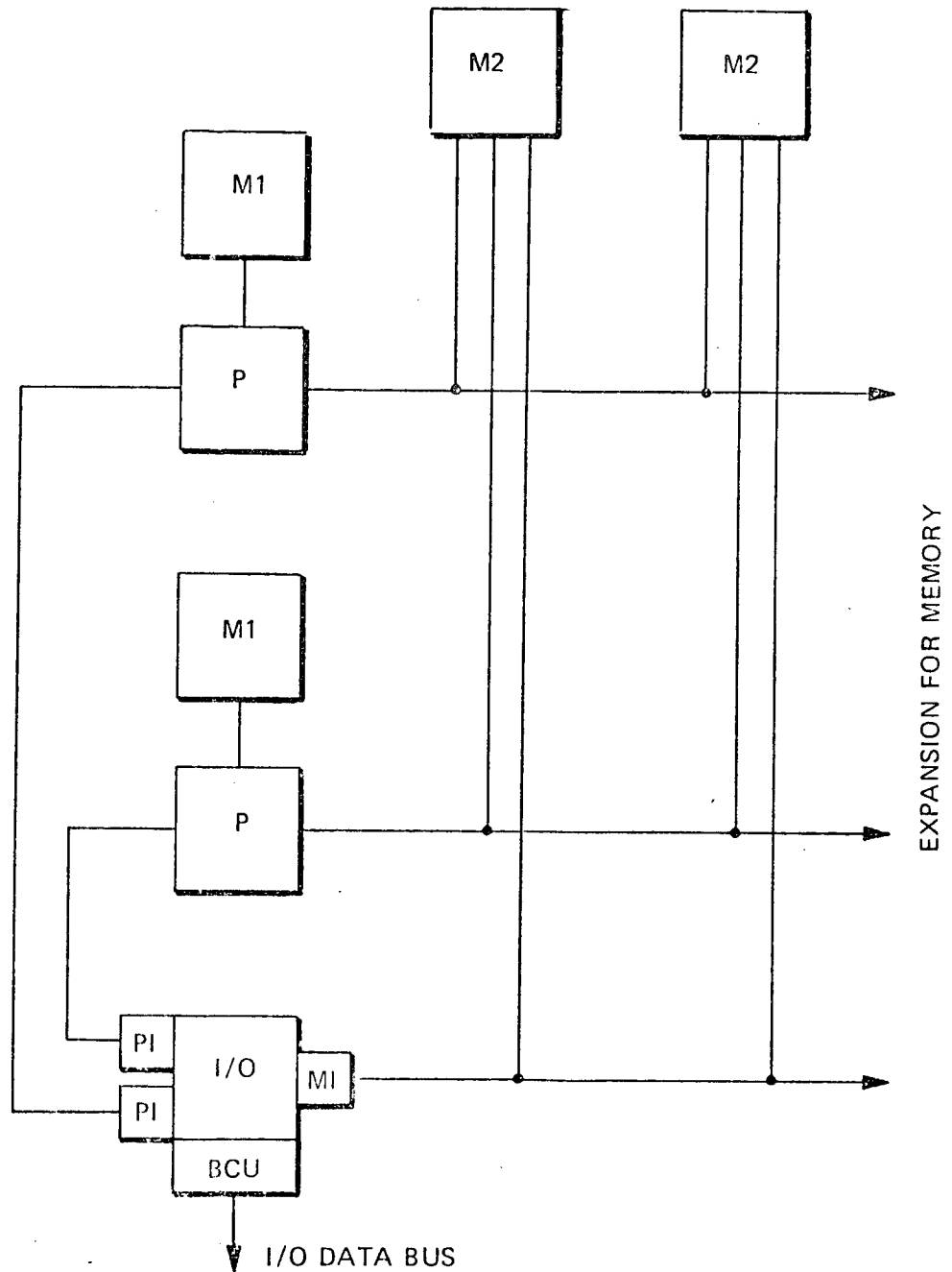
Figure 7-1 presents a simplex version of the proposed configuration. The purpose of this diagram is to illustrate the basic elements of the multiprocessor. A dedicated bus multiport memory configuration is illustrated for the internal bus. Also shown are separate paths between the P and I/O units. The utilization of dedicated buses is not that critical in the configuration, and the principles to be discussed could be implemented with a time-shared bus or even a cross bar switching mechanism.

The interface between the P and I/O units is conceived to consist of two signals, one directed from P to I/O and the other from I/O to P. The signal initiated by P and sent to I/O, tells the I/O unit to indirectly fetch through a fixed M2 location control information concerning an I/O command. Data transfer to or from the data bus always goes directly to M2. The signal initiated by the I/O and sent to P is an I/O interrupt and instructs the P unit to look indirectly through another fixed M2 location to ascertain what the I/O unit wants. The type of information that the I/O communicates to the P unit includes command execution completion and I/O unit, data bus or peripheral unit failure indications.

7.1.2.1 Error Detection and/or Correction

Two types of error must be considered. These are errors due to transients and errors due to permanent hardware failures. Transient errors are generally not caused by a hardware failure but rather by a source of external noise. It is therefore impossible to isolate the source of the error by testing since the hardware operates satisfactorily. The state of the hardware is altered by the transient and the valid state is restored. It is therefore not necessary for a spare unit to be switched in order to perform recovery. Any successful error detecting technique must detect both transient and hardware failures. A number of methodologies may be applied to provide the error detection capability.

7.1.2.1.1 Periodic Diagnosis. This method relies on software to periodically initiate test sequences and compare the results with predetermined values. For a number of reasons, periodic diagnosis is unsatisfactory for error detection within the multiprocessor. First, there is no guarantee that the error is detected before damage has been caused in terms of bad write operations



BCU - BUS CONTROL UNIT FOR DATA BUS
 MI - I/O MEMORY INTERFACE
 PI - PROCESSOR I/O INTERFACE

Figure 7-1. Simplex Multiprocessor Schematic

into M2 or incorrect I/O commands. Second, there are failure modes which can prevent a processor from sequencing, requiring, therefore, some form of hardware time-out to be provided. Third, the categorization of all the possible failure modes and the execution of the tests to interrogate the possible failures can be a significant software effort (experience has shown this can amount to more than 50 percent of the total). Fourth, there is a large probability that transient errors will not be detected in time for satisfactory recovery.

7.1.2.1.2 Error Detecting Codes. The commonly applied parity coding is an effective method of detecting single errors, but is ineffective in protecting against transients which affect more than a single bit. This is the statistical independent requirement.

Coding techniques have been studied for many years. Codes may be classified as either transmission codes or arithmetic codes. Both may be used for error detection on the internal bus and in memory. Transmission codes do not retain their error detection characteristics under arithmetic operations. As a matter of fact, neither transmission nor arithmetic codes retain their characteristics under nonlinear binary operations.

There are a number of points to consider before relying solely upon coding for error and failure detection.

1. Not all component failures result in erroneous data .
2. The cost of the additional hardware may be more effectively applied to other techniques of detection and correction.
3. Extensive error coding imposes higher bit rates, and degrades performance.

7.1.2.1.3 Component Level Redundancy. This methodology applies redundancy at the circuit or gate level. It is possible to design fail safe logical systems, in the sense that the system continues satisfactory operation even if a component fails or a single bit transient error occurs. By incorporating redundancy as an inherent part of the initial design procedure, rather than after the design is accomplished, logic can be synthesized which is tolerant of single component failure.

Quadded logic allows a number of logic gate failures within a digital computer without disturbing its capacity to perform the function for which it was designed. This technique can, in theory, increase reliability by orders of magnitude.

Another form of logic level redundancy is known as interwoven redundant logic. With this technique, each gate receives a number of versions of each input and forms its output from the redundant input information. Certain redundant gates, while performing logic, can correct errors from the previous stage in one layer of a multilayer structure. Other redundant gates correct errors in two alternating layers. Majority voting and quadding techniques are special cases of interwoven redundant logic.

All of the above techniques suffer from three major drawbacks:

1. On the basis of gate count alone, the application of component level redundancy is expensive (at least four times the simplex version)
2. The large increase in the number of interconnections between the redundant gates can produce unreliability
3. It is a very difficult problem to maintain statistical independence between failures when one considers problems associated with mechanical packaging and power supplies

7.1.2.1.4 Functional Level Redundancy. Functional level error detection is at the level of the arithmetic unit, operating registers, shifter, etc. If the functional units are synchronized properly, then the output nodes of the redundant units can be compared and errors detected. Even at this level, there can exist single failures that prevent error detection; for example, a power supply transient.

Functional units need not necessarily be 100-percent duplicated to provide error detection. Parity bits may be added to each logical word of the nonredundant system. However, parity will not detect all single component failures.

Majority voting is a technique where each functional unit or system is triplicated and the output is chosen to agree with the majority of the individual outputs from the triplicated elements. An example of an application of majority vote techniques is the Saturn V computer. Adaptive vote taking is a modification and extension of the majority vote technique. This method employs more than three versions of a functional unit output. When one unit fails, it is automatically switched out of the majority vote network, allowing a greater increase in reliability over conventional majority vote techniques.

7.1.2.1.5 Modular Redundancy. A question arises as to whether functional level redundancy with comparators at each functional intersection or modular redundancy (a complete duplicate P, M2 or I/O unit) provides better error detection properties. From a system performance point of view it is necessary to detect the inability of a subsystem to communicate correctly with its environment. A duplicate module operating independently of the first with a comparator to detect any difference in outputs is just as effective in detecting errors as functional redundancy within the module, provided that error sources are statistically independent in the two modules. In fact, modular redundancy may very well be more cost effective, since the total component count will not, depending of course on the complexity of the comparator, approach that of the functional redundancy examples in the previous section.

7.1.2.2 Fault Isolation

Once a failure is detected it must be isolated to the level of a replaceable unit so that recovery can begin. In the case of majority voting, fault

isolation can be performed by the hardware. It is not a mandatory step since error correction is automatically assured. In the case of two identical redundant units, a comparator cannot tell which unit is bad. It can only indicate a discrepancy and it is necessary to invoke further diagnostics, usually via software, to check the results to determine which unit is bad. Software, of course, requires time for execution and can never be fully effective. In a multiprocessor it is possible for the P elements to perform self-test in addition to cross-checking between two P elements. For M2 units or I/O units the P element initiates all fault isolation test sequences. A philosophical question arises as to whether it is possible to verify that the particular fault isolation software can isolate all possible fault situations. Experience with automated test equipment indicates that this is a formidable task.

7.1.2.3 Failure Anticipation and Recovery

When a failure is detected and isolated, enough valid information must survive the failure to enable a spare resource to be initialized to the correct state so that recovery and continued operation can be effected. The logic mechanism required in anticipation of recovery can be very complex, and must be thoroughly designed if it is to prove cost effective.

MIT has proposed the concept of a single instruction restart (SIR). The concept was developed to make recovery from hardware failures and transients transparent to the programmer. A fundamental tenet in the SIR concept is that all errors are detected within the instruction in which they first occur, so there is no propagation of errors to subsequent instructions. Each instruction is divided into two parts; a compute phase and a store phase. During the compute phase, results are stored in a temporary buffer and during the store phase they are transferred to their final storage location. Each phase is made restartable. When an error is detected the instruction phase in which the error occurred can be automatically re-executed.

The main motivation for SIR was the Apollo experience in which restart points were maintained dynamically during execution of the programs. With a random error rate of 1 in 10^{12} for the Apollo guidance computer (AGC), an error might occur within several hundred hours. This error rate imposed a requirement for a means of recovery after a transient error is detected. For each program module a restart pointer was maintained so that a preplanned sequence could be executed in case of an error. The development and testing of this mechanism was extremely costly. It is estimated to have consumed over 50 percent of the total software effort. It should be realized that the AGC was a simplex computer with no hardware redundancy. When an error was detected, action had to be taken immediately. That is, the recovery program could be initiated at any point in the middle of any program sequence.

On the other hand, if hardware redundancy, for example, majority voting, is employed a program module can be allowed to continue to the end before fault isolation and recovery are initiated. This will relieve the programmer of many of the problems associated with restart, since he may conceive of the situation as one in which errors can only occur at the end of program modules and not in the middle of critical sequences.

7.1.2.4 Tolerance in P

Error detection within the P element of the multiprocessor is most important. A permanent or transient failure can give rise to a number of different error patterns. Some of the problems in the detection of these error patterns are discussed below.

7.1.2.4.1 Problem Areas.

1. Fan Out

In the process of logical design, a single gate is often used to drive many different gates. This is called "fan out". The failure of the driving gate not only causes its output to be incorrect, it can also propagate to the output of all the gates to which it is attached. An example where this problem directly affects the data content within a P element is in data busing.

Internal to P are a number of buses or common points where many different data registers may be gated. The failure of a logic element or connection in the data bit logic usually manifests itself as a single bit error. This type error can be easily detected by conventional parity techniques. However, the failure of a control line used to gate information onto the bus can manifest itself in a multiple error situation. In the worst case, all the bits on the bus could be in error.

If two registers are gated onto the bus at the same time, a logical OR between the two registers occurs. No mechanism exists which can, in general, detect this situation short of a complete duplicate bus with redundant control lines.

In terms of coding theory, if there are N information bits, a minimum of N check bits must be used to detect a burst of N errors. This busing example shows a situation in which a single component failure caused a potential N bit burst error.

2. Arithmetic Unit

In discussing the failure tolerance aspects of the AU, one must distinguish between the coded form of the data inputs and the actual information which the coded form represents. The coded information may possess redundant information which can be exploited to provide error detection.

An AU may be defined as having two inputs, an operation and an output. A and B are the information inputs, C is the information output and OP is the operation defined between A and B:

$$A \text{ OP } B = C,$$

OP is typically ADD, SUB, Logical AND, Logical OR, Exclusive OR, MULT, DIV, etc.

Now, if $F(A)$, $F(B)$, and $F(C)$ are the coded words, there must exist an operator $*$ such that,

$$F(A) * F(B) = F(C)$$

Also, to be useful, in error detection, the function F must contain enough redundant states to detect all single errors. One may consider two types of F functions. Separate codes enable $F(A)$ to be the juxtaposition of A and some redundant check bits $G(A)$ with a check bit operation $*$:

$$F(A) = A, G(A)$$

$$A \text{ OP } B = C \qquad G(A) * G(B) = G(C)$$

Nonseparate codes do not lend themselves to this partitioning.

Literature exists which deals with the problem of arithmetic codes. That is, $OP = \text{ADD or SUB}$. Both separate and nonseparate codes exist. However, if $OP = \text{Logical AND or Logical OR}$, there is conjecture that no efficient separate code exists. To be efficient, the number of bits necessary to represent $F(A)$, while still retaining a single error detection capability, must be less than twice the number of bits necessary to represent A .

It is not our purpose to prove the mechanisms of arithmetic codes. Suffice it to say that they exist. All the arithmetic codes provide error detection by performing a residue check on the result of the arithmetic operation.

The utilization of a nonseparate code creates problems for the programmer in performing nonarithmetic functions. An example of a nonseparate arithmetic does is one in which every number is multiplied by 3. After an addition, the result, taken modulo 3, must be zero or an error has occurred.

One of the most efficient separate codes is where $G(A)$ is the parity function of A . However, the logic necessary to compute the parity function of the sum of two numbers requires the duplication of the carry logic. In addition only half of the possible single component failures are detected.

The main point in this discussion is that any coding method used for error detection within an AU is quite complex in terms of the number of logic gates.

3. Encoding and Decoding

Within the logical structure of P are areas in which bit patterns must be decoded or encoded into different formats. The decoding may or may not produce mutually exclusive outputs. These processes may, in many instances, allow a single component failure at the input to the combinational structure to produce multiple errors on the output of the combinational circuit.

Address decoding is an example in which parity can help in detecting errors. Associated with each decoded state is an input parity state. A single error on the input of an address decoder will change the parity bit state. This fact can be exploited in error detection. However, the failure of the address decoder in such a manner as to select two addresses simultaneously is very difficult to detect.

4. Control Unit

The utilization of a microprogrammed control memory greatly reduces the problem of error detection. Parity associated with each control word is very effective. However, accessing the wrong control word because of a failure in the addressing logic can cause catastrophic results.

Incorrect addressing can be caused by incorrect address decoding, which could result in the selection of the incorrect control word or the logical OR of two control words.

5. Packaging and New Technology

Before presenting possible solutions to the problem of error detection a few points will be made considering the technology with which the next generation system will be built.

- a. The cost of a system will be a function of the number of different types of circuit packages, and the external connection complexity (pin count) of each package, and more or less independent of the complexity within a given package.
- b. The number of kinds of functions must be minimized. This will lead to greater use of programmable functions utilizing read only memories.
- c. The interconnections between LSI circuits must be minimized to achieve a high level of reliability.
- d. Functions such as arithmetic units will be integrated on one LSI chip.

As an example of the cost of an error detecting code, consider the case of a register-to-register transfer. Information transfer from one register to another can be protected by the attachment of a parity bit. Considering present day MSI technology an 8 bit register is packaged on one chip. Similarly, an 8 bit parity generator or checker is also packaged on the chip. Therefore, the parity logic constitutes just as many circuits as the register itself. This seems to require twice the logic.

To obtain independent failures the error detection logic and the processing logic must be packaged separately. This implies that they cannot be manufactured on the same LSI chip. If error detection is accomplished at the level of the register, or arithmetic unit, then the error detection logic will consist of at least the same number of LSI packages as the functional logic.

One may consider the use of redundant registers with a comparator. This would require three times the logic.

7.1.2.4.2 Recommendations and Conclusions

1. It will require at least twice the logic, and subsequently at least twice the cost to detect all possible single component failures in P.
2. Redundancy at the processor level seems the most practical because:
 - a. The redundant processors can be packaged separately with independent power distribution logic. This will more closely approach the failure independence assumption.
 - b. Redundancy with a comparator at only one interface will reduce the number of interconnections between the redundant processors.
 - c. Redundancy at the module level can yield a degree of reconfigurability. If the failed half of the duplex processor is switched out, the reliable half can still be used by itself or in conjunction with another processor.
 - d. Error detecting codes won't detect all possible errors.
 - e. Periodic software self-test won't catch all failures before they propagate to multiple errors.
3. To reduce the problem of fault isolation and to implement rapid or instantaneous recovery a triple processor with majority voter is recommended.

4. Since it is assumed some functions may require error detection and switchover within tens of microseconds, a triple processor with voter is almost mandatory.
5. The voting element must be redundant to prevent the nonreporting of failures.
6. The internal structure of each P element can be simplified since no error detecting hardware need be incorporated. The entire error reporting mechanism is contained within the voter element.

7.1.2.5 Failure Tolerant Operating Memory, M2

The criticality requirements indicate that redundancy of M2 for recovery need only require at most 12.5-percent more memory. This assumes that all single component failures can be detected.

The error detection could possibly be accomplished with a completely redundant M2 with a comparator. However, this is very costly. A method will be described in which complete error detection with M2 can be accomplished without a doubling of memory.

It is assumed that the processor contains local storage, M1, and transfers between M1 and M2 are multiword (block oriented).

7.1.2.5.1 Error Detection Criteria. The major problem in applying error detecting codes to memory storage is that of assuring that a single component failure does not cause undetectable multiple errors. To meet the Fail Operational requirements all component failures must be detectable before the failure propagates. For the purpose of this discussion, the following errors must be detectable for a memory error detection proposal to be satisfactory.

1. All bit failures within a word must be detectable
2. Addressing an incorrect word must be detectable
3. Addressing an incorrect block must be detectable
4. The failure of a memory module to sequence must be detectable
5. Errors yielding an all "0" or all "1" word must be detectable

In conventional 2, 2-1/2 or 3-D core or plated wire memory systems all these errors can result from single component failures.

7.1.2.5.2 Proposed M2 System Properties. To achieve the above five goals an M2 configuration with particular properties is proposed. The interaction between M2 and P plays a large role in the error detection process. The configuration possesses the following aspects.



1. The entire operational memory, M2, possesses a partial backup M2', which is sufficient to aid in recovery of critical tasks. If the allowable recovery times for time critical tasks are sufficiently long for critical programs to be read from M3 into an alternate M2 module (i.e., greater than 10 milliseconds), M2 need only serve as storage for the redundant write operation, less than 12K words. If recovery times are more stringent than this, invariant critical programs will need to reside in M2' in addition to the variables. M2' may then exceed 30K words.
2. The block transfers between M1 and M2 allow the application of a block code. This code (shown in Figure 7-2) utilizes one parity bit per word and one parity word per block. If addressing to a level less than a block is desired the entire block is still accessed to provide error detection.

If a block of information words contains B bits per word and W words per block, then a total of $W + B + 1$ extra bits are required per block of $W \times B$ bits.

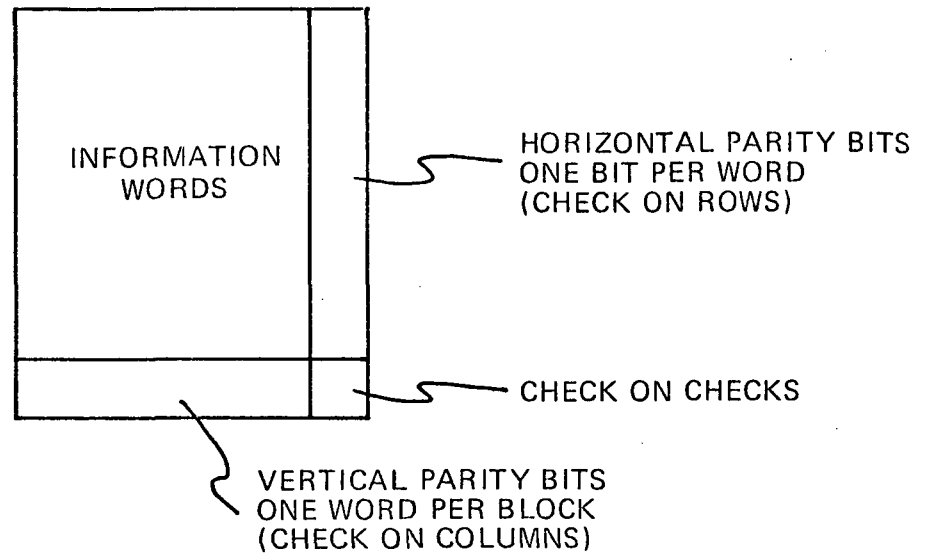
For a 16-word block of 32 bit words, containing a total of 512 bits, an aggregate of 49 extra bits would be required.

3. The parity check bits, both vertical and horizontal, will be stored in both M2 and M2'.
4. All check bits are verified after each read operation. The P unit always reads M2 for information and M2 and M2' for the check bits. The interface of the P unit performs the parity verification automatically in the hardware as well as compares the check bits.
5. The interface between P and M2 is physically separate from the interface between P and M2'. That is, P has two buses to memory, one for M2 and one for M2'.
6. The P element will contain a time-out mechanism to indicate that the memory has not responded within a predetermined time limit.
7. The first word in each block will contain the block address.

Let us now examine how well the above properties satisfy the five error detection criteria presented in Section 7.1.2.5.1.

1. All single bit errors are detected by both horizontal parity bits and vertical parity words. As a matter of fact, single bit errors are correctable although this feature of the block code is not exploited. Two dimensional odd parity will not detect symmetrically clustered even numbers of errors in different words as below.

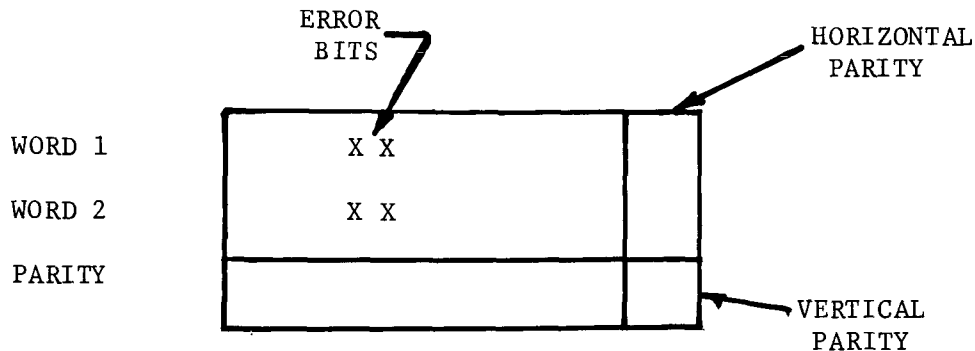
GENERAL ARRANGEMENT



EXAMPLE - ODD PARITY, 5 BIT WORD, 6 WORD BLOCK

1	0	0	0	1	1
0	1	1	1	0	0
1	0	1	1	1	1
1	1	1	1	1	0
0	0	0	1	1	1
1	0	1	0	1	0
1	1	1	1	0	0

Figure 7-2. Block Code for M2



The probability of this occurring in practice can be minimized by careful memory design. Logically adjacent bits can be assigned physically nonadjacent memory arrays by bit-plane organization, interleaving, etc.

2. If M2 accesses an incorrect word, the failure can be detected by two mechanisms. The horizontal parity will detect the error in one-half the time. The main mechanism, however, is the vertical parity word, which will indicate a multiple error. If the vertical parity word does not detect the failure, then the accessed word must have had the same data content as the desired word. This is, of course, perfectly satisfactory.
3. If an addressing failure causes the access of an incorrect block in M2, then the vertical parity word comparison will indicate the failure. Similarly, the horizontal parity bits will also make the indication positive.
4. The improper sequencing of a memory module can manifest itself as an incorrect word. This is detectable by parity checks. The nonsequencing of M2 or M2' results in a memory hang-up situation. This must be detected by the time-out mechanism in the requesting P or I/O element.
5. Burst errors which cause all the bits in a memory word to become all "0" are detected by the horizontal odd parity bits and vertical parity word. An all "1" burst error in a word is detected by the vertical parity word.

It seems from these considerations that all failures in M2 modules can be detected by the above scheme. Failure in the parity generation and error detection logic is a failure in the P or I/O interface. A failure in the P or I/O interface is discussed later. A precise determination as to whether the block oriented M2 with check bits in M2' will satisfy the 100-percent error detection criteria must depend upon a careful analysis of a specific M2 design and technology. This detailed analysis is beyond the scope of this present effort.

7.1.2.5.3 Memory Failure Recovery. For time critical (TC) functions which contain program and data in redundant storage, M2', recovery from a memory failure is a simple matter. Since the P unit detects the failure during a read block operation, the P unit need only initiate a command to read the redundant copy from M2'.

In those cases where an addressing failure during the addressing of TC information causes a different block to be read from M2 and M2', a procedure must be initiated to isolate the bad block. This can be accomplished by comparing the address contained within the first word of the block with the desired address. Depending upon recovery time requirements for TC functions this procedure may be accomplished in software, firmware, or hardware.

In case of a transient which destroys the contents, but not the mechanism of M2, backup programs from M3 must be read in and allocated memory space. This can proceed while critical tasks are still being executed out of M2'. Complete restoration of M2 including critical and noncritical tasks might take hundreds of milliseconds. However, as long as M2' can execute the TC functions the situation is satisfactory.

7.1.2.6 Fault Tolerant Input/Output

It is not specified at this time the maximum recovery time from failure in the I/O unit. The resolution of this question is the same as for P. It is a matter of time criticality. As with P, two I/O units are required for failure detection. It is assumed that software fault isolation will require too much time. Therefore, a triple redundant I/O unit is proposed.

7.1.3 Summary of Central Multiprocessor Design Factors

The major controlling factor in the configuration is the failure tolerance requirements. This is closely followed in importance by those factors which tend to reduce the software cost by making the system easier to program and test. It is probable that the software cost for the space station will ultimately overshadow the cost for the computer hardware. Therefore, the judicious incorporation of hardware which relieves the programmer of implementation details related to machine functions can create a more cost effective system.

7.1.3.1 Failure Tolerance Features

1. Each processor element will be triply redundant with adaptive voting elements.
2. Each I/O unit will be triply redundant with adaptive voting element.
3. Memory error detection will be provided through exploitation of a block code with check bits and information bits being stored in independent memory modules

4. M2 memory will have an area in which critical functions operate so that fail operational performance can be attained for critical tasks.
5. The internal bus must contain dual redundancy so that a failure in a bus interconnection does not cause overall system failure.

7.1.3.2 Memory Hierarchy

A number of different levels of memory must be considered. Numbers presented below are only tentative.

1. M0. This is the microprogrammed control memory. The speed of M0 will probably be in the 50 to 100 ns, access time range. The number of bits is a function of the number of micro controlled macros desired for a given performance level. Probably a total of 25,000 bits would prove satisfactory. This could be arranged in an appropriate combination of micro and nano memories, taking advantage of both horizontal and vertical micro-programming features.

At least part of M0 should be RAM memory. We suggest at least 10,000 bits.

2. M1. This is the local storage dedicated to each P element. Its speed is in the 100 to 500 ns access time range. If used as a cache, 4K to 8K of 32 bit words would be very adequate. Cache memory size used with the IBM 360/85 was determined experimentally by executing various mixes of machine code.

It is not clear at this time what the size of or even the need for M1 will be in the case where the P element executes instructions at a higher level than the base 360 set from which currently reported M1 performance specifications were derived. This is a function of the locality of HOL instruction flow and HOL data. This subject will be investigated later.

3. M2. This is the operational memory. The speed of this memory is in the 1 to 2 microsecond range. Although it was indicated in Section 2.0 that 90K of memory is required with an appropriate memory management algorithm for communicating between M3 and M2, either via paging or segmentation, the requirements imposed upon M2 can be reduced to the 23K to 64K range, with 32 bit words organized into blocks of 16 to 64 words each. The size of the backup module M2' is determined by (a) the recovery time constraints of critical tasks, and (b) storage for the vertical parity bits associated with the blocks. The bit total is directly proportional to the number of stored blocks, which is inversely proportional to the number of words in a block. For 16 words in a block the total number of parity bits approaches 10 percent of M2 storage. For a 64-word block the total number of parity bits is less than 5 percent of M2

storage. The minimum size for M2' is approximately 4K, 32 bit words. For this situation 64 word blocks are assumed. Time criticality is such that all backup is contained in M3. On the other extreme, M2' can be 32K, 32 bit words. This is arrived at by assuming 16 word blocks and that all critical programs and data are contained in M2'. For the purpose of this discussion, M2' is considered to be the totality of all backup operational memory.

4. M3. This is the mass memory. It has a latency time of the order of 1 to 10 ms. All but the high iteration rate functions reside in M3. Any expansion of DPA functional capability will probably occur in terms of 600K, 32 bit words are required for M3.
5. M4. The archival memory is large, consisting of many millions of words. It has been defined not to impact design factors within the central multiprocessor.

7.1.3.3 Processing Element

1. The P element will contain a local memory M1 to reduce bus traffic and increase processing speed.
2. Microprogrammed control will be utilized.
3. Hardware will be incorporated to aid in implementing a higher order language. This includes the hardware control of stack mechanisms, the automatic setting and testing of descriptors, and the direct execution of a reverse polish string or intermediate level language.

These hardware elements tend to make performance criteria such as MIPS or EAPS less meaningful than for machine language execution. The decision to store instructions in terms of the more semantically concise HOL notation will tend to reduce the amount of M2 and M3 storage required.

7.1.3.4 Input/Output Unit

1. The I/O unit will be microprogrammed controlled. The memory can be ROM except during the development stages of the system, when a RAM will make for easier system integration.
2. A maskable interrupt between the I/O unit and the P elements will be provided. Details concerning the interrupt will be placed in dedicated M2 locations by the I/O processor. It is better to provide the interrupt capability at this initial stage of planning, and to disable it by means of a mask, than to determine a later requirement for it and to have no way of implementing it.

3. The I/O unit should be allowed an independent asynchronous interface with P, since it would lead to a less complex, and therefore, reliable design.

7.1.3.5 Internal Bus

The choice of configuration and performance criteria for the internal bus is a subject to be covered later. A decision is not given at this time. However, a few observations can be made.

1. Enough capability should be provided so that the bus is not the limiting factor in performance, especially in the maximally expanded configuration.
2. The number of interconnections between communicating elements should be minimized consistent with performance factors.

7.1.3.6 Memory Protection

1. The M2 complex will be provided with logic to automatically control the interlock mechanism as well as to prevent and abort deadlocks.
2. The hardware testing of descriptors for memory protection will be provided.

7.1.3.7 Some Operational Considerations

1. The transfer mechanism between M1 and M2 will probably consist of a hardware controlled algorithm with a limited associative memory to aid the search procedure.
2. Transfer control between M2 and M3 will probably consist of a combination of paging and segmentation. Again, some special associative hardware, packaged within the M2 complex, will aid in the dynamic allocation and deallocation of M2 physical space.
3. Transfer control between M3 and M4 is not a major time factor. It can be controlled entirely by software. M4 can be program controlled like any external I/O unit. It is not considered to be part of the M1-M2-M3 virtual memory system.

7.2 HOLD MEMORY ORGANIZATION AND INTERNAL BUS DESIGN

The remainder of Section 7.0 develops the concept of a higher order language machine (HOLM). In order to study the merits of a HOLM and its possible application in the central multiprocessor, the memory hierarchy and the internal bus design were studied. A functional division of the CP memory into the M1, M2 and M3 levels (defined in Section 7.1) is made. Recommendations of internal organization, performance, and technology for each level are presented. An analysis of bus traffic in a HOLM environment 13 is given. Problems of conflict and implementation are discussed. A recommended bus transfer concept is described.

7.2.1 Higher Order Languages and Higher Order Language Machines

This paragraph addresses the topic of employing a higher order language for the central multiprocessor. Before proceeding with the detailed discussion some clarifications must be made concerning the programming language and the machine upon which the language is executed. To aid in the understanding of the differences between a higher order language and a machine-oriented language, the following definitions are presented.

1. Higher Order Language (HOL). An HOL is a notational system which allows the programmer to define the problem to be solved in a form which reflects the characteristics of the problem, and which is related to the programmers natural mode of expression.
2. Machine-Oriented Language (MOL). A MOL is a notational system which enables the programmer to describe his problem in terms of the instructions of the available computer. Since most present day computers are based upon the Von Neumann type architecture a MOL possesses executable statements such as LOAD, STORE, ADD, SUB, CONDITIONAL BRANCH, etc.
3. Higher Order Language Machine (HOLM). A HOLM is a machine which directly executes HOL statements. There is no need to convert the HOL into some MOL by means of a preliminary compilation in order to process the HOL statements. In other words, a HOLM is a piece of hardware which was designed with the characteristic structure of a particular HOL in mind and which therefore executes the basic HOL operations efficiently.
4. Machine-Oriented Language Machine (MOLM). A MOLM is a conventional machine whose basic instructions are biased toward the control and manipulation of data by the basic elements of its architecture.

The relationship between the language and machine is depicted in Figure 7-3. If a MOLM is used to execute a HOL, then a translation is required to convert the HOL into the MOL. This process is usually referred to as compilation. If the HOL and MOLM possess a large mismatch the inefficiencies can occur in terms of speed of execution and utilization of memory.

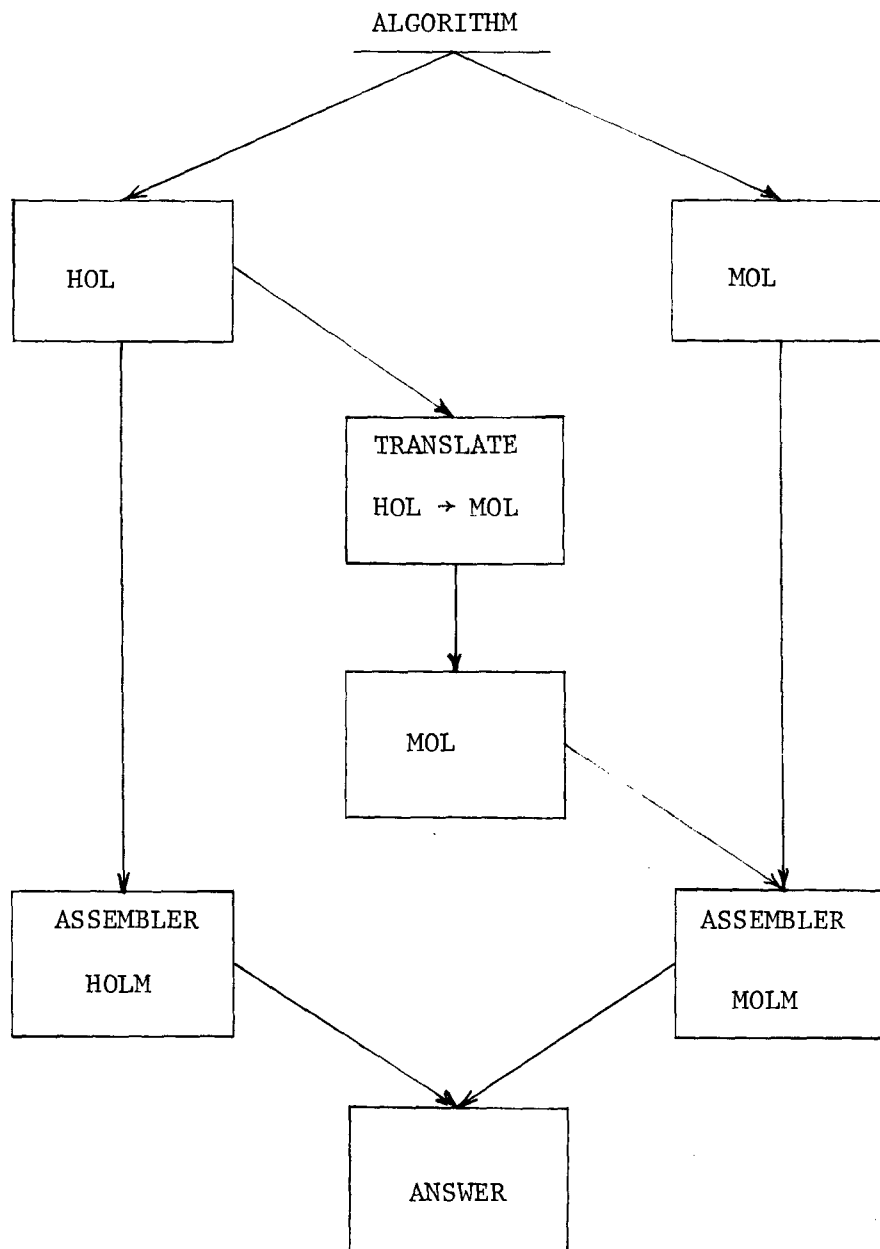


Figure 7-3. Language and Machine



The assembly process takes the MOL or HOL, converts operations to binary form and establishes the relationships between symbolic addresses and physical memory space. For the purpose of this discussion the assembler is an intrinsic part of the machine.

7.2.1.1 Justification for Using a HOL

There are a number of standard arguments in favor of using a HOL over a MOL as the basic language of a programming effort (3, 4, 5)*.

1. Ease of Communication within the Program

- a. The program becomes self-documenting and therefore reduces the cost and need for separate documentation for different levels of management (e.g., mission definition, analysis, program specification).
- b. The ability to communicate allows the analyst who develops the equations to program them. This avoids the inherent difficulties of communication that occur between differently oriented groups of people.
- c. In any large project, the problems of maintainability are aggravated by the inevitable turnover of personnel. Not only must different people be able to maintain the program, but they must also be able to easily modify, add, or redesign sections of the software.

2. The HOL is chosen because it is oriented to the problem being solved and uses language more natural to the programmer. The concise formulation of the problem is therefore enabled. This leads to:

- a. Fewer errors due to conceptual difficulties and the different ways of stating a problem.
- b. Shortened program design and development time.

3. The programmers need be less concerned with the following traditional machine features and problems:

- a. Scaling and precision problems
- b. Base register allocations
- c. General register considerations
- d. Initialization problems, particularly in loops
- e. Data protection

4. The HOL allows program transferability from one machine to another, eases debugging, reduces checkout problems due to problem oriented modularity and separation from hardware.

* References are listed at the end of the section.

5. Carey and Sturm (Reference 6) present some interesting facts concerning the costs of existing space software and the projected cost savings of a compiler for aerospace programming. In particular, they are concerned with the SPL compiler. The following information is extracted from the above reference to indicate the software cost for aerospace missions:
- a. The cost of software for manned space missions is two to four times the hardware cost.
 - b. The Apollo Saturn V's instrument unit software was produced at a rate of 2.5 instructions per man-day.
 - c. As much as 1 to 2 months was needed to make a 500 to 1000 instruction change in the Titan III computer.
 - d. Software checkout is very expensive and not perfect. A single error in a 2000 instruction space program might require 50 to 100 validation runs on a simulated ground-based machine. Extrapolation to a 25,000 instruction program indicates 1000 to 1200 runs.
 - e. Typically, 100 instructions in new unvalidated machine code written by a senior programmer may contain 3 to 8 errors. Carey and Sturm estimate 10 to 70 percent of these errors can be avoided by the use of a compiler.
 - f. By hand, machine code typically is produced at a rate of 270 to 350 instructions per man-month. With a compiler, 500 to 540 instructions per man-month is possible.
 - g. Writing a JOVIAL compiler for an IBM 4 Pi computer would cost between \$300,000 and \$500,000.

Software is indeed expensive. To quote Carey and Sturm,

"But software is soft more in name than in fact. The dollars involved are hard, and so are delays in software development."

6. Lest the cost of the compiler frighten anyone, consideration must be given to the alternative; namely, the generation of an assembler. IBM (7) states that two to five man-years were required to produce assemblers for various space-qualified computers. The assembler for the B-70 computer required three to four man-years, the Gemini computers took two to three man-years. The 4 Pi/CP required four to five man-years. A second assembler for the CP required two additional man-years. If we estimate 40 to 50 thousand dollars, loaded, per man-year, the cost of an assembler is the \$100,000 to \$250,000 price range. In the same reference, IBM states that it was cost-effective to produce Algebraic Language Translator (ALT) compiler just

for the purpose of reducing programming errors. This compiler was developed for the TC-2 computer for the A-7D/E aircraft even though it was not part of the contract.

7.2.1.2 Justification for Using a HOLM

Computational inefficiencies occur whenever a compiler is required to translate the statements of the problem into machine instructions due to the mismatch between the computer architecture and the HOL architecture. The design of a machine which matches the language will not only eliminate the processing inefficiencies and improve performance over a conventionally structured MOLM, but it will also reduce memory requirements because a HOL statement is more semantically concise and economic of space. A number of designs have been proposed and implemented. The following estimates of performance and cost reductions have been made.

1. Kerner and Gellman (Reference 2) have designed a machine which directly executes Fortran statements. Programs written in this language and executed on their machine occupied 75-percent less memory. This conclusion was reached by comparing the machine code generated by the Fortran compiler for the IBM 7094 with the number of words required to represent the instructions for the HOLM. The 4:1 compression of memory space for program storage was the result.

Since a significant portion of the memory in the space station is required for storage of programs, a substantial cost savings is possible. An estimate of the logic cost for the Fortran language processor (FLP) was made. It was estimated to require 45,000 gates, which is equivalent to a medium- to large-scale computer.

An interesting analysis was conducted considering a computer aboard and earth-orbiting space station. The computer was estimated to contain 260,000 32-bit words. Sixty-thousand words were allocated to executive program data and work area, leaving 200,000 words for instruction storage. The instruction compression ratio of 4:1 was applied indicating a memory savings of 150,000 words or 5.7 million bits.

The analysis indicates, that considering the logic necessary to implement the FLP, the memory and power supply, a savings of 315 pounds could be obtained. At a launch cost of \$1000 per pound, this amounted to \$315,000 per vehicle.

2. Sugimoto (Reference 1) has studied the direct execution of the PL/1 language. He has actually implemented the PL/1 reducer, and has some experimental results. For typical scientific programs, the length of the object code has been reduced by 25 percent compared to the object code generated by presently available PL/1 compilers. He also found a speed gain of 28 percent for arithmetic string operations.

It should be emphasized that Sugimoto's data are experimental, while Kerner and Gellman's work is analytical and the result of simulation tests.

3. Rice and Smith (Reference 8) discuss the SYMBOL system which includes the development of a language and hardware to execute the language. They concluded that the direct hardware implementation of a general-purpose, high-level language and a good conversation mode system can save up to 50 percent of the overall facility cost of a good conventional system using a general-purpose, high-level, batch-oriented system.

7.2.2 Memory Hierarchy

The different levels of memory hierarchy were defined in paragraph 7.1.3.2. Figure 7-4 illustrates four of these levels and the data paths between them. Memories M1 and M2 will be discussed in the following paragraphs; the mass memory and the archival memories are discussed in Section 8.0.

7.2.2.1 Design of M1

Five categories of local functions can be identified:

1. High-speed buffer for interface to M2
2. HOL instruction buffer
3. Data buffer and search memory
 - a. Descriptor cache
 - b. Value cache
4. Control stack
5. Evaluation stack

7.2.2.1.1 Interface Buffer. This buffer accepts blocks of five words which arrive from M2 at intervals of 200 microseconds per word. Transmissions are checked for parity and content and are dispatched to the other M1 areas for processing. The size of this buffer need not exceed a few words. Its speed must be commensurate with the delivery rate of data (i.e., less than 200 ns cycle time).

7.2.2.1.2 HOL Instruction Buffer. This buffer holds the stream of HOL particles, probably one byte each, while they are decoded and executed. The size of this buffer is determined by the degree of locality in the instruction stream, and the value of instruction hit ratio desired. A buffer size sufficient to hold small routines would aid in keeping program references local. A 64 or 128 word buffer is suggested. A buffer that is more than one block large immediately incurs the problems of mapping from M2, determination of presence and location of desired information, and replacement

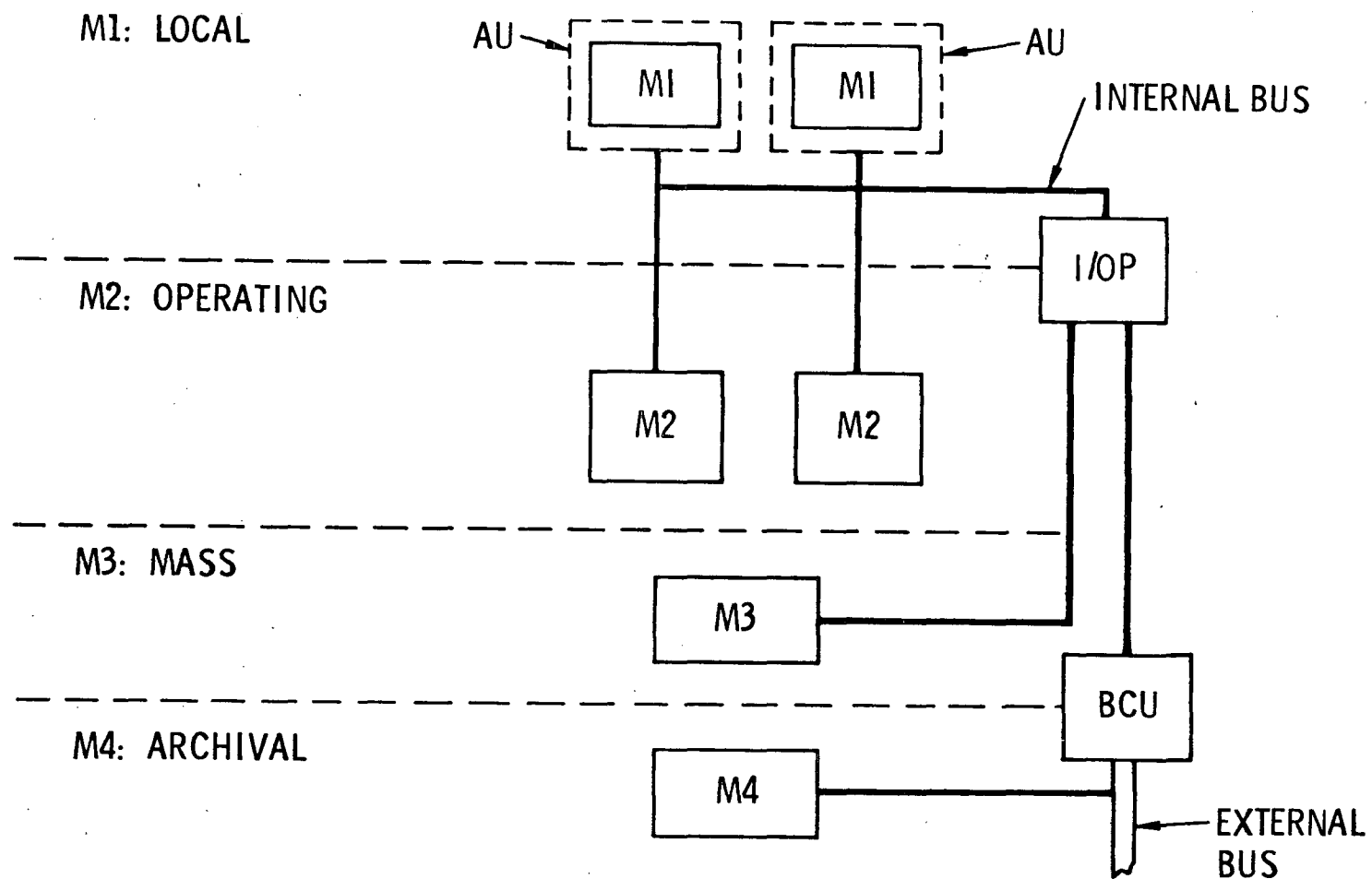


Figure 7-4. Memory Hierarchy

policies. A size of 64 words implies a 16 level associative search for the desired block (4 words per block); this is well within current state of the art, even at 100 ns to 200 ns cycle times.

7.2.2.1.3 Data Buffer. The use of descriptors as a control and protection mechanism is proposed. The execution of a HOL statement is expected to generate several times as many references to data elements for the purpose of manipulation in control stacks, etc., as for actual numerical evaluation. Hence, it is proposed to provide separate storage and control for descriptors and operand values. Sumner (Reference 10), reporting on the design of the MU5 computer of Manchester University, found in practice that most programs contained less than 100 names; individual routines, less than 64. Accordingly, a 64-level associative store for procedure and data descriptors is proposed.

Storage for operand values should be sufficient to contain the larger data structures which the HOL will recognize. A degree of associative search in the value buffer may assist the evaluation of repetitive operations involving large structures, although this may only be determined by closer analysis of real situations. An operand value store of about 64 words is proposed with, at this time, a random access organization.

7.2.2.1.4 Stack Mechanisms. It is proposed to assign a control and evaluation stack mechanism to each individual AU. A stack will help to improve the hit ratio by maximizing local operations. The exact implementation of the stack is not important: most stack mechanizations involve a conventional memory address and control organization with stacking being handled by pointers or indirect addressing. Special purpose hardware using shift registers may yield higher performance. Although control and evaluation stacks are conceptually separate, it is possible to combine them in a single mechanisms, as in the SPL machine of Keeler, et al (Reference 9).

This study does not allow a detailed enough analysis to establish specific organization, size and speed requirements for the proposed stack memory. However, for completeness, it is estimated that each stack (control and evaluation) should be about 64 words deep. Stack overflows must be provided for by continuing each stack in appropriate regions of M2.

7.2.2.1.5 M1 Summary. We have just outlined five major functional areas to be considered as the sum total of the local buffer memory M1. Each has been very approximately sized at 64 words as illustrated in Figure 7-5. The total M1 capacity does not exceed 350 words, or about 10^4 bits. Consequently, the constraints on the technology to be used for M1 are considerably eased. It is very likely, in fact, that the proposed M1 will prove smaller and considerably less difficult to design than the micro and nano memories required in each AU to implement the HOL execution capability.

The overall speed requirement in M1 is difficult to assess in view of the uncertainty concerning HOL operation. It can certainly be stated that it will not exceed that required to sustain the 10^6 equivalent add instructions per second of a conventional architecture (i.e., about 500 ns cycle time), although certain portions, such as the M2 interface buffer, must operate at less than 200 ns for reasons stated above. Current technologies

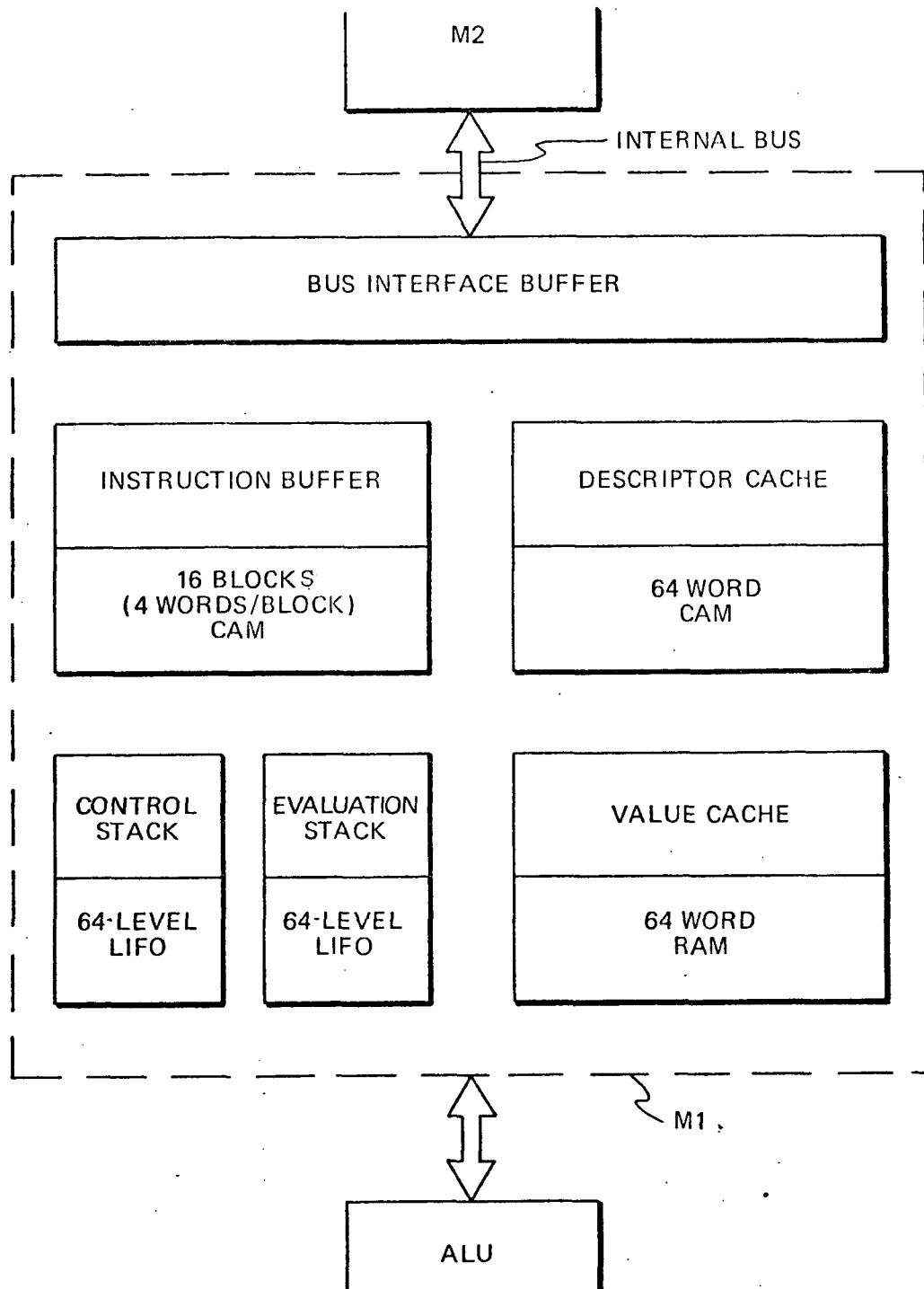


Figure 7-5. M1 Functions

Storage for operand values should be sufficient to contain the larger data structures which the HOL will recognize. A degree of associative search in the value buffer may assist the evaluation of repetitive operations involving large structures, although this may only be determined by closer analysis of real situations. An operand value store of about 64 words is proposed, with, at this time, a random access organization.

7.2.2.1.4 Stack Mechanisms

It is proposed to assign a control and evaluation stack mechanism to each individual AU. A stack will help to improve the hit ratio by maximizing local operations. The exact implementation of the stack is not important: most stack mechanizations involve a conventional memory address and control organization with stacking being handled by pointers or indirect addressing. Special purpose hardware using shift registers may yield higher performance. Although control and evaluation stacks are conceptually separate, it is possible to combine them in a single mechanism, as in the SPL machine of Keeler, et al. (9).

This study does not allow a detailed enough analysis to establish specific organization, size and speed requirements for the proposed stack memory. However, for completeness, it is estimated that each stack (control and evaluation) should be about 64 words deep. Stack overflows must be provided for by continuing each stack in appropriate regions of M2.

7.2.2.1.5 M1 Summary

We have just outlined five major functional areas to be considered as the sum total of the local buffer memory M1. Each has been very approximately sized at 64 words, as illustrated in Figure 7-5. The total M1 capacity does not exceed 350 words, or about 10^4 bits. Consequently the constraints on the technology to be used for M1 are considerably eased. It is very likely, in fact, that the proposed M1 will prove smaller and considerably less difficult to design than the micro and nano memories required in each AU to implement the HOL execution capability.

The overall speed requirement in M1 is difficult to assess, in view of the uncertainty concerning HOL operation. It can certainly be stated that it will not exceed that required to sustain the 10^6 equivalent add instructions per second of a conventional architecture (i.e., about 500 ns cycle time), although certain portions, such as the M2 interface buffer must operate at less than 200 ns for reasons stated above. Current technologies available to implement a 10^4 bit memory in this speed range include plated wire and bipolar, MOS and CMOS semiconductor LSI. A choice of specific technology must fall on the secondary factors of cost, power dissipation, weight and volume. Another factor to be considered is that a semiconductor approach allows logic and memory functions to be combined which provides it a considerable advantage in the design of associative and/or search memories. The complementary MOS technology has extremely attractive properties: high speed (<50 ns), low power (10^{-9} watts/bit), and noise immunity. It is apparently developing

available to implement a 10^4 bit memory in this speed range include plated wire and bipolar, MOS and CMOS semiconductor LSI. A choice of specific technology must fall on the secondary factors of cost, power dissipation, weight and volume. Another factor to be considered is that a semiconductor approach allows logic and memory functions to be combined which provides it a considerable advantage in the design of associative and/or search memories. The complementary MOS technology has extremely attractive properties: high speed (50 ns), low power (10^{-9} watts/bit), and noise immunity. It is apparently developing strongly in the commercial field, which is always an auger that a technique is about to become generally accepted, and by post-1975 CMOS may be the overwhelming candidate for space station use. If an M1 technology must be pin-pointed today, it would be CMOS.

In summary, the M1 functional and performance specification is as follows:

1. Organization

- a. Interface buffer. 5 to 10 32-bit words; 200 ns cycle time word-by-word shift register at input, 4-word parallel transfer at output
- b. Instruction buffer. 64 words arranged in 16 four-word blocks, with each block containing a search tag to enable 16-level associative search for desired block
- c. Data buffer. One 64-word content-addressable memory (CAM), and one 64-word location-addressable (random access) memory
- d. Control stack. Two 64-word last-in, first-out lists; probably arranged as location-addressable RAM with address pointers

2. Performance

Apart from interface buffer section, which must cycle at 200 ns interval, 200 ns to 500 ns cycle times for all sections.

3. Technology

For post-1975: CMOS. Bipolar LSI and plated wire would be strong back-up candidates.

7.2.2.2 M2 Design and Interface to M3

This paragraph will be mainly concerned with the capacity requirements of the operating memory M2, the organization of the data, and the management of the flow of program and data between M2 and M1, and between M2 and M3. The M2 access speed has already been largely determined by the discussion of M1 in the previous paragraph. In order to make appropriate design decisions we must preview some of the results of the paragraphs that follow. The discussion on failure tolerance concludes that:

1. Two physically independent M2 complexes are required to provide recovery for critical functions in the case of an M2 module failure. Critical variables are stored redundantly in each complex. Critical instructions and fixed data are permanently resident in M2, but are not redundantly stored in M3.
2. Each M2 complex consists of five physically separate modules, four for program and data and the fifth for storage of the check word with which all information transfers are validated.
3. The mass memory M3 is used as the primary storage medium only for noncritical programs and data. For this reason, no redundancy of the M3 unit is proposed.

The total M2 storage requirement for the M2 design proposed in this report can be summarized as follows. The estimated differences between a conventional and a HOL machine are shown for comparison.

Total M2 Storage Requirement (32-bit words)

<u>Function</u>	<u>Conventional</u>	<u>HOL</u>
Critical instructions	24,254	8,040
Critical fixed data	5,523	5,523
Critical variables	6,523	6,523
Critical variable redundancy	6,523	6,523
Overlay area	<u>32,768</u>	<u>16,384</u>
Total information storage	75,591	42,993
Error checking (+25 percent)	<u>18,898</u>	<u>10,749</u>
Total storage	94,489	53,742
200% design/growth margin	<u>188,978</u>	<u>107,484</u>
Grand total with maximum expansion	283,467	161,226

The following comments are made concerning the breakdown of M2:

1. A HOL approach is expected to require only 30 percent of the storage for instruction of a conventional machine.
2. No storage reduction for data in a HOL machine is assumed.
3. These estimates reflect the functional breakdown of Section 2.0 which identified a total storage requirement amounting to 90K words. The subsequent storage requirement of 67K words was not functionally broken down, making it impossible to estimate the individual functions. The figures are therefore very conservative.

3. The overlay area is the region of M2 reserved for executions of noncritical programs permanently resident in M3. It contains both instruction and data area. The instruction storage area is reduced for the HOL for reasons of semantic conciseness. The data areas are not modified from original estimates.

The above total M2 requirement is split between the two M2 complexes. For each complex the sizing is as follows, assuming a HOL organization.

Required Storage for Each M2 Complex

<u>Function</u>	<u>Number of 32-Bit Words</u>
Critical instructions and fixed data (half in each M2 complex)	6,787
Critical variables (redundant storage in each M2 complex)	6,523
Overly area (half in each M2 complex)	8,192
Error checking (half in each M2 complex)	<u>5,375</u>
Total basis	26,877
Additional 200% (for design and growth margin)	<u>53,746</u>
Maximum capacity per M2 complex	80,623

Since each M2 complex consists of five separate modules, no module need exceed 16,384 words, even in the maximum configuration.

Information is referenced and transferred from M2 to M1 in five-word blocks. It is therefore natural to control its storage and management within the M2 complex in a block-oriented fashion. The speed requirements indicate that a new word from M2 is to be delivered to the bus every 200 ns. This can be achieved without imposing a difficult constraint on the M2 technology by interleaving a number of modules of more modest performance. For the five-day interleaved M2 complex, the information is organized as follows:

	<u>Module 1</u>	<u>Module 2</u>	<u>Module 3</u>	<u>Module 4</u>	<u>Module 5</u>
Block 1	Word 1	Word 2	Word 3	Word 4	Word 5
Block 2	Word 1	Word 2	Word 3	Word 4	Word 5
.
.
.
Block n	Word 1	Word 2	Word 3	Word 4	Word 5

The major properties of this arrangement are:

1. Addressing of each M2 complex need only be to the level of a block. A reference to any word in the block by the ALU results in the whole block being accessed.

2. The words of which each block is composed are stored in physically independent modules, which eliminates most common mode sources of error against which the two-dimensional parity error detecting scheme might be ineffective.
3. Expansion within each M2 complex is achieved by adding further blocks. This can be accomplished by substituting larger modules rather than by adding modules. This is deemed to be less difficult in terms of the bus interface logic and the address sequence, and memory control logic unique to each M2 complex. The individual modules of M2, sized at a maximum of 16K words, can initially be implemented at less than 8K, and expanded as additional memory requirements develop during the MSS lifetime (Note, however, that it is essential to size the block addressing scheme to the maximum capabilities from the outset).

A block diagram of the elements of a five-way interleaved M2 complex is shown in Figure 7-6. Each module is complete with its individual address and control logic and memory buffer register. The master controller and sequencer (MSC) manages the scheduling of the individual modules, and resolves the conflict arising from simultaneous requests to one M2 complex from the three processing elements; Aul, Au2, and the I/O.

The AU and I/O interface constitute the three ports into each M2 complex. Each interface element performs the following functions:

1. Bus interface
2. Data verification
3. Command decoder
4. Block address buffer
5. Data buffer

Another outcome of resolving conflict on the block rather than the word basis is a reduction in the average block transfer rates experienced by each AU. This is because with block resolution the MSC is capable only of processing one block at a time. If only one AU is accessing memory, then it will receive or transmit words at the rate of five words per M2 block cycle. If two AU's are accessing memory at the same time each receives or transmits words at an average rate of five words per two M2 block cycles. That is, the M2 complex handles the words at its maximum rate independent of the number of requests.

The utilization of M2 is proposed as follows:

1. Those programs that have been identified as critical are to be resident in M2. They are to be assigned fixed predetermined and permanent areas in the M2 memory map. Specific locations can be assigned (i.e., the mapping of name space into physical M2 space will be done once, for critical routines, at compile time). No overlapping of critical programs upon each other, or of folding

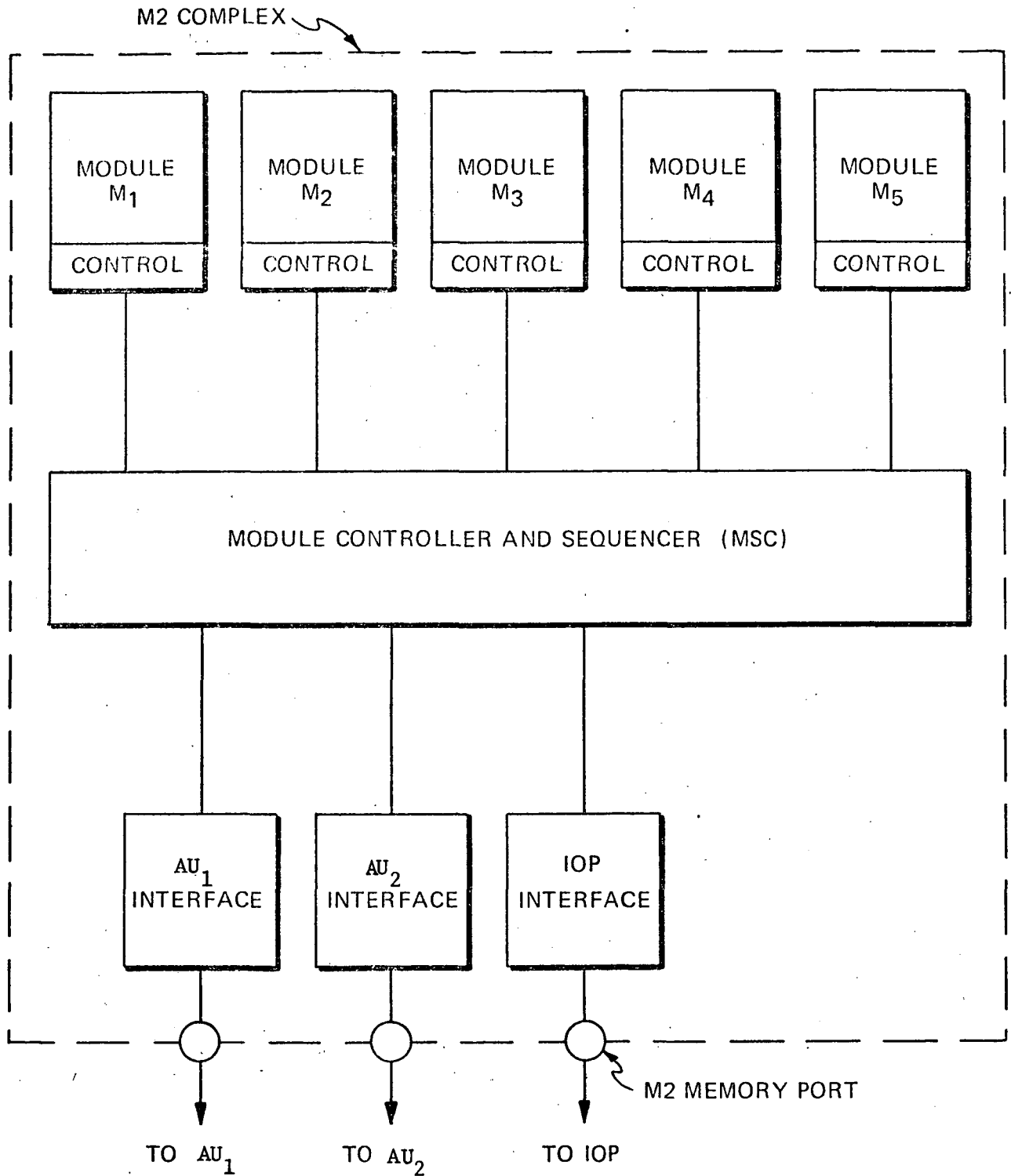


Figure 7-6. 5-way Interleaved M2 Complex

upon themselves, will be allowed. The unique location of critical software in M2 will enable the recovery of critical functions to be more easily accomplished.

2. Programs corresponding to noncritical functions will be permanently resident in M3. As they become active they will be called by the executive into an area of M2, which has previously been referred to as the "overlay area".

The overlay area constitutes "multiplexing" of M2, because a finite memory space is used to satisfy the storage demands of programs whose sum total may exceed what is available by a considerable amount.

Many of the noncritical functions are very mission-mode dependent. That is, they are executed at their specified iteration rate, only during specific mission times. For example, all the shuttle alignment, terminal rendezvous and docking function need only be read into the overlay area when the shuttle is docking with the space station. Functions associated with crew medical functions, station scheduling or solar array control are not required during this mode.

None of the noncritical functions which are required at all times have iteration rates higher than once per second. All these functions can be segregated into overlay sections in M3 and be prescheduled to be overlayed into M2 in a fixed sequence. This allows a degree of look-ahead in accessing M3 in anticipation of execution.

The degree of M2 time sharing which can be achieved can only be determined by a closer analysis of the individual timeline of the noncritical functions. The impact of multiprogramming (processor time sharing) should be considered in parallel with M2 overlaying to effect an optimum M2 utilization. It has been assumed that an M2 region of 16K words will satisfy the instantaneous requirement of any set of active noncritical functions.

Technology is not a constraining factor in the design of M2. The requirements for an internal M2 module are a maximum capacity of 16K words, random accessible with one microsecond cycle time. 16K 32-bit words amount to a half million bits. A module of this capacity with a one microsecond cycle time presents no problem to the current technologies of ferrite core, plated wire or MOS. The choice must, as in the case of M1, rest on secondary factors. Plated wire is the strongest contender, being nonvolatile, non-destructive readout, and capable of low power operations.

7.2.3 Internal Bus

It has been determined that the internal bus is essentially a memory switching and control mechanism as shown in Figure 7-7. It enables the AU's and I/O units to communicate with the operating memory M2. Two signals are postulated between an AU and I/O. The signal directed from I/O to AU is a maskable interrupt and tells the AU to look into M2 to interrogate the nature of the interrupt. The signal directed from AU to I/O causes the I/O to look into M2 to obtain information concerning the nature of the I/O procedure to be executed.

Figure 7-7 also shows that the mass memory M3 possesses a separate input into the I/O element. The I/O unit not only controls information on the data bus but also the flow of traffic between M2 and M3.

Several bus configurations were evaluated to arrive at a final configuration. Detailed design factors were discussed and a tradeoff evaluation presented. In order to perform this evaluation a number of ground rules were established to arrive at a consistent result.

First, it should be realized that any bus configuration can be designed to meet any particular communication requirement. The variables such as bus width, speed per wire, power dissipation can be varied and different technologies applied to meet the requirements. If different implementation technologies are employed for each configuration, the detailed designs must be performed before a valid comparison of the total cost can be made. This is clearly beyond the level of effort planned under the current contract.

The different configurations have been evaluated on the basis of the following three ground rules.

1. All configurations were postulated to provide the same level of performance. That is, each will allow enough traffic so the AU's can achieve their specified instruction rate and the I/O can achieve its through-put rate.
2. The same technology was applied to each configuration. This indicates that for a given bus length the maximum bit rate that an individual wire may sustain is the same for each configuration.
3. For evaluation purposes, fault tolerance aspects were not considered at this time. Redundancy was considered after a basic architectural philosophy was established.

The bus configurations considered are depicted in Figure 7-8.

7.2.3.1 Internal Bus Traffic Requirements

Requirements imposed upon the internal bus possess two aspects. The first is the derivation of the total number of bits per second the bus must sustain to meet the instruction rate requirements. The second aspect deals

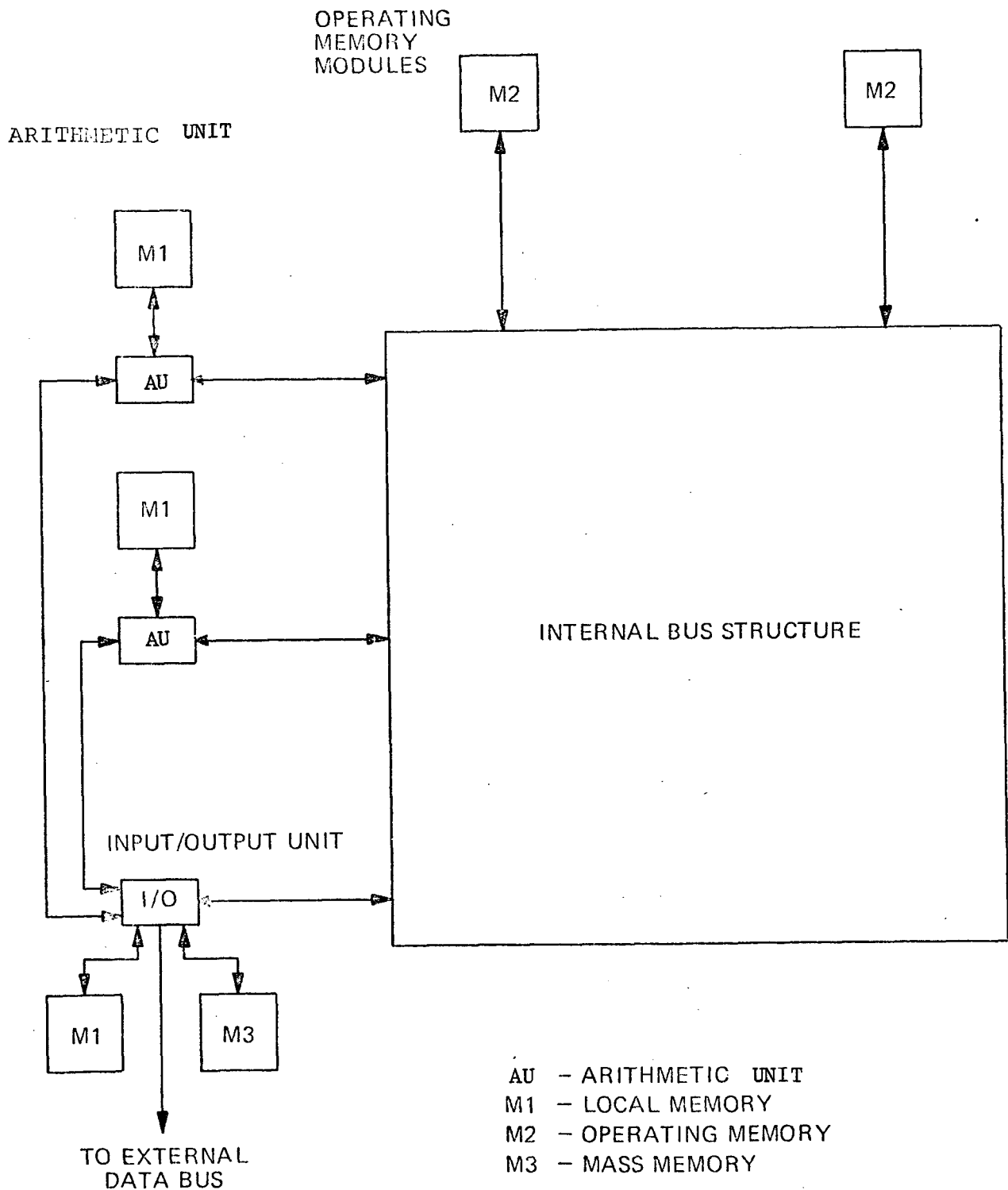
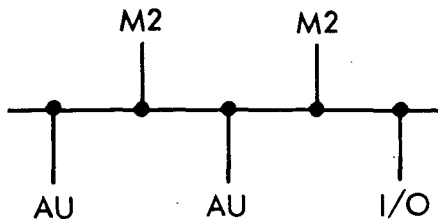
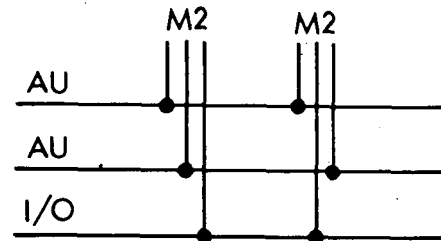


Figure 7-7. Internal Bus

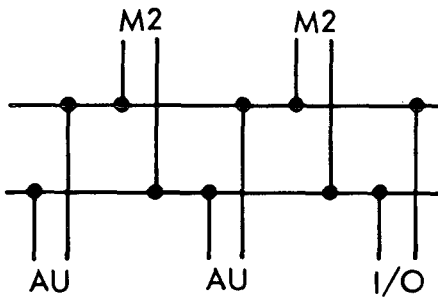
CONFIGURATION 1
SINGLE TIME MULTIPLEXED BUS



CONFIGURATION 3
DEDICATED BUSES WITH
MULTIPOINT M2



CONFIGURATION 2
MULTIPLE TIME SHARED BUSES
WITH MULTIPOINT ELEMENTS



LEGEND:


 PHYSICAL
 CONNECTION

Figure 7-8. Internal Bus Configurations

with the physical packaging of the system and determines the bit rate limit which may be expected from a given technology.

The total bit rate which the bus must carry may be calculated by considering the total number of instructions per second which the AU's must execute and the traffic imposed by the I/O unit including data bus traffic and M3-M2 data transfer.

The following analysis will take the instruction rate requirement and translate them into bus rate requirements for both a conventional 360 type computer structure, and for an AU element which contains M1 and executes higher order language statements directly from a reverse polish stack.

From Table 2-2, the maximum requirements imposed upon processing speed is 1.893 MIPS. These instructions relate to a 360 type computer organization.

Analysis of two different sets, each of approximately 200K instructions, shows an average of 7.2 bytes fetched from memory per instruction executed. Hence, the bit rate is close to $8 \times 7.2 = 57.6$ bits per instruction, exclusive of parity, in these samples.

Using this number, we find that dynamic instruction execution with a 360 type architecture imposes a traffic rate of $1.893 \times 57.6 \text{ Mbps} = 110 \text{ Mbps}$.

In addition to the 110 Mbps, the bus must carry the addressing and control bits necessary to initiate a memory request. In a word organized M2 this would require one address word per fetched word (each fetched word being 32 bits). If 18 bits are assumed to be sufficient for addressing each 32 bit word, then the bus must sustain an additional $18/32 = 56.4$ percent of the original 110 Mbps, for a total of 62 Mbps. This results in a total bit rate of $110 + 62 = 172 \text{ Mbps}$ for a conventional computer organization without a local M1 memory and a word addressed M2.

If the HOLM organization proposed in Section 7.2.1 is incorporated, a reduction in bus traffic can result. This organization assumes a specialized M1, 4-way interleaved M2, with direct storage and execution of reverse polish HOL instructions.

In a dynamic execution situation the above analysis of instructions indicates that from 47 percent to 50 percent of the memory accesses are for operands while the remainder are for instruction. Using the average 48.5 percent for operands, we conclude that out of a total 110 Mbps, 53.5 Mbps are associated with operands and 56.5 Mbps are associated with instructions.

In processing a HOLM instruction set, the number of bits of instruction information that need be processed will be substantially reduced. We shall assume that 70 percent of the 56.5 Mbps associated with instruction fetches can be saved. This yields a reduction of 39.6 Mbps.

As in the previous discussion of the assumed 360 organization, addressing and control will add to the overall bus traffic requirements. Assume 16 bits are required to address and specify a 4-word (128 bit) transfer. Then, $16/128 = 12.5$ -percent more bits must be added.

Table 7-2 summarizes these conclusions and indicates a total internal bus traffic due to instruction of more than a factor of two, as compared to a conventional structure. Assuming two AU's, then 40 Mbps will be considered to be the bus traffic created by each AU. As a side issue parity bits and check words should be considered.

Table 7-2. Internal Bus Traffic Due to Instruction Execution

	Conventional Structure	HOLM Structure
(Maximum instruction rate)	(1.893 Mips)	
Bit rate due to instructions	56.5 Mbps	
Bit rate associated with operands	53.5 Mbps	
Bit rate due to addressing	62.0 Mbps	
Total conventional	172.0 Mbps	
Reduction due to HOL semantic conciseness		-39.6 Mbps
Subtotal of instructions and operands		110 - 39.6 = 70.4 Mbps
Addressing (+12.5% of subtotal)		9.9 Mbps
Total HOL machine bus traffic		79.3 Mbps

The 40 Mbps rate imposed by each AU represents an average required to meet the performance specification. The peak rate is limited by one of two factors: the first is the speed at which the M2 complex can provide or absorb information. The second is the speed at which M1 can provide or absorb information. A 4-way interleaved M2 complex with a 1 microsecond cycle time per internal module can supply a peak data rate of 128 Mbps. A 5-way, 1 microsecond, interleaved M2 complex corresponds to 160 Mbps. Similarly, an M1 memory with a 250 ns cycle time and a 32 bit word corresponds to 128 Mbps and a 200 ns M1 corresponds to the 160 Mbps peak rate.

Figure 7-7 indicates that the I/O unit contributes to the internal bus traffic for three reasons: the external data bus, instruction execution, and the M3-M2 transfer mechanism.

The maximum design limit imposed upon the external data bus is 10 Mbps. In the worst-case situation all these bits will appear on the internal bus. This factor can, however, be reduced by considering the impact of the data bus control units (DBCUs) and the I/O unit in compressing the internal bus bit rate. The DBCU generates certain addressing and control information

automatically upon a trigger command from the I/O unit. This type of traffic does not originate from M2 and should be subtracted from the 10 Mbps. An analysis of the data bus for the shuttle indicates that for messages of length 10 bytes, 50 percent of the data bus traffic is associated with addressing and control. This, therefore, reduces the impact on the internal bus to 5 Mbps.

If we assume that the I/O unit is structured like a processor, then the instruction execution will add additional bus traffic. The I/O processor will require a smaller instruction set (compared to the AU) which will be compatible with the more limited language used to program the I/O functions. Because of the very specific tasks (data flow control) assigned to the I/O, program locality should be very high. Much higher than for the AU SDC's IMSIM simulation indicates that the I/O unit accounts for about 12.5 percent of all instructions that are executed. This would create bus traffic of 10 Mbps for a HOLM and 25 Mbps for a conventional structure. This impact can be further reduced by an estimated 50-percent due to the very high anticipated temporal locality. In a HOLM, I/O traffic, therefore, only contributes 5 Mbps while in a conventional structure I/O contributes approximately 11 Mbps.

A mechanism for M2-M3 control will produce additional bus traffic. The peak rate is determined by the access rate of M3. A drum will yield about 6 Mbps. A plated wire mass memory will produce approximately the same rate if power dissipation is to be kept to a low level. Combining the data bus, instruction execution and M3 transfer mechanisms a total of 16 Mbps is imposed by the I/O unit for a HOLM, and 21 Mbps for the conventional structure.

Table 7-3 shows the average bus traffic requirements for the three major internal bus configurations. Comparisons between a conventional AU structure and a HOLM is also made.

7.2.3.2 Internal Bus Design Factors

The following paragraphs present a discussion of the various factors which must be considered in specifying the internal bus.

7.2.3.2.1 Synchronization. One of the first factors is the synchronization of the interfaces to the bus. If complete phase synchronization can be accomplished then the logic required to control and interface to the internal bus can be minimized. A completely phase-locked system does not impose an insuperable technological problem. The computer display channel (Reference 11) synchronizes the system at 7.5 MHz with equipment separated by more than thirty feet. No major technical problems were encountered during the design or during operation. Phase synchronized internal bus interfaces are, therefore, proposed.

7.2.3.2.2 Speed Per Wire. The following analysis will indicate the maximum bit rate that can be sustained per wire. The following assumptions are made concerning the analysis.

1. The bus drivers and receivers should use the same integrated technology utilized in the design of the other logical elements of the system.

Table 7-3. Bus Traffic Summary for Configurations

Configuration 1--Single Time Multiplexed Bus Configuration 2--Double Time Multiplexed Bus Configuration 3--Dedicated bus			
	Configuration		
	1	2	3
<u>Conventional Structure</u>			
Traffic on each time multiplexed bus	193 Mbps ⁽¹⁾	96.5 Mbps ⁽²⁾	--
Traffic on bus dedicated to AU	--	--	86 Mbps ⁽³⁾
Traffic on bus dedicated to I/O	--	--	21 Mbps ⁽⁴⁾
<u>HOLM Structure</u> ⁽⁵⁾			
Traffic on each time multiplexed bus	96 Mbps	48 Mbps	--
Traffic on bus dedicated to AU	--	--	40 Mbps
Traffic on bus dedicated to I/O	--	--	16 Mbps
NOTES: (1) The total traffic load of 172 Mbps (Table 7-2) for instructions plus 21 Mbps for I/O is carried on the time multiplexed bus. (2) The double time multiplexed bus carried one-half of the 193 Mbps on each leg of the bus (on the average). (3) Each AU sustains one-half of the 172 Mbps required for instructions. (4) See paragraph 7.2.3.1 for rationale (5) Estimates for HOLM follow the same philosophy as for a conventional structure			

2. Very conservative design ratings in terms of delay and speed will be used to maximize component reliability.

3. The internal bus structure should be as simple as possible. This implies no pipelining in the bus. That is, the bus is allowed to settle to its quiescent voltage level before a new transfer is initiated.

Figure 7-9 indicates the basic elements of the internal bus interface. This figure indicates the total delay accumulated in transmitting one bit over a bus wire. The numbers were obtained by the following reasoning:

1. The flip-flop delay is obtained by considering a J-K flip flop such as an SN 54105.



2. The delay of the line driver was estimated by considering an AN55110 line driver. Assuming 10 pf per foot of cable, a total of 150 pf capacitive load was postulated.
3. The delay of the bus wire was estimated by postulating 1.5 ns/foot propagation velocity.
4. The line receiver delay is typical of an SN55108 line receiver.
5. Clock skew is an estimate of aging of components once the system has been initially adjusted.

If the communications elements utilize a clock width of 20 ns, then for T²L flip flops signals cannot change at their inputs within this 20 ns. Therefore, total time between bits on a wire is the 150 ns delay plus 20 ns dead time which equals 170 ns. This yields a maximum bit rate on a wire of 5.85 Mbps. A more conservative figure, which will be used, is 5 Mbps per wire.

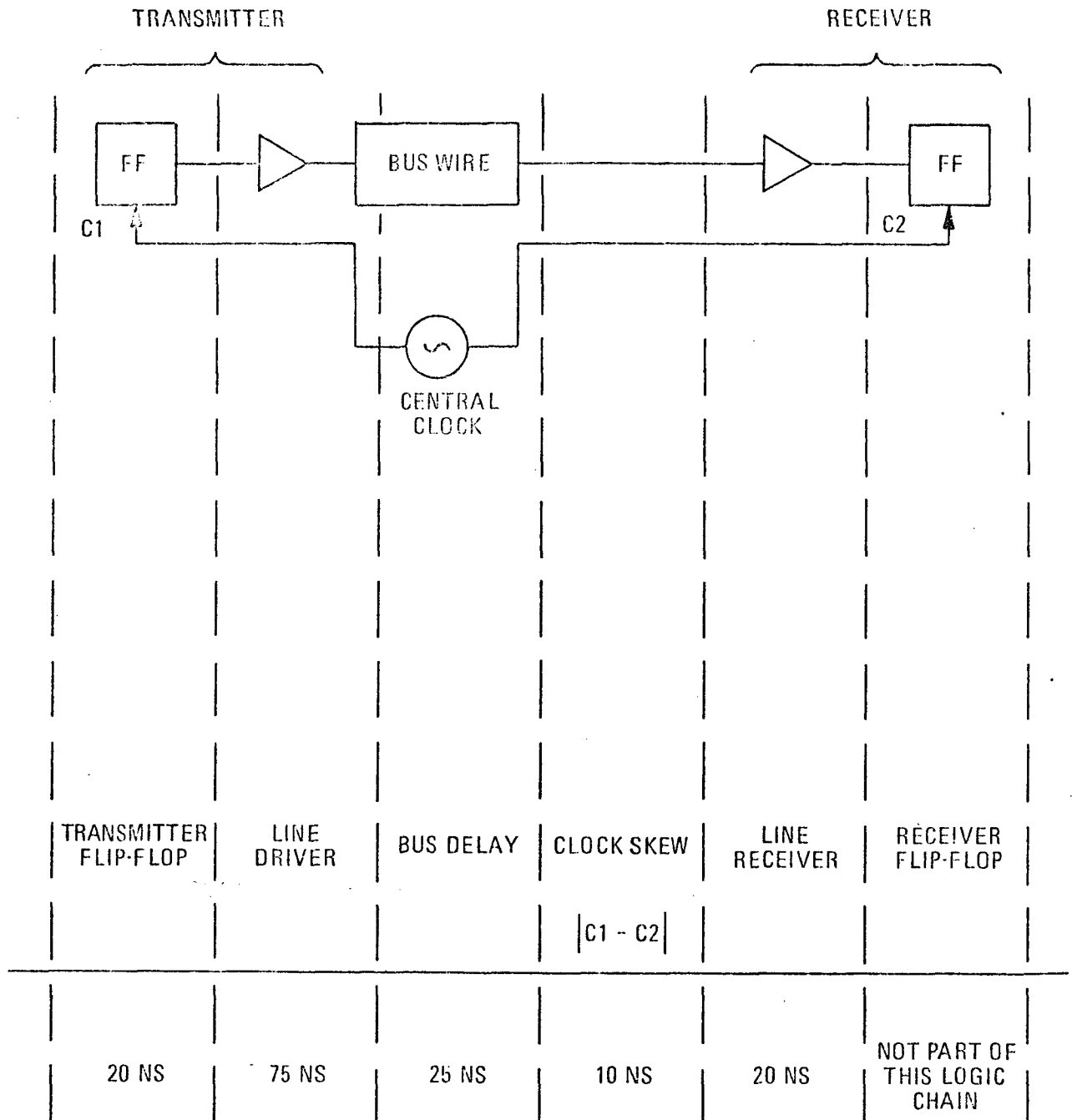
7.2.3.2.3 Number of Bus Wire. Unfortunately, the internal bus must be designed to handle the peak traffic load. For a 32 bit wide data path and 5 Mbps, a total maximum transfer rate of 160 Mbps can be achieved. This is exactly compatible with the 5-way interleaved M2 memory complex. Since the bus can handle 5.85 Mbps, the utilization of 5 Mbps on each wire is clearly conservative.

7.2.3.2.4 Bus and Memory Conflict. For a time multiplexed bus, one must consider conflict over the use of both the internal bus and the memory. For a dedicated bus system only memory conflict need be considered.

For the purpose of the following discussion the parity bits and check word transmitted with each block will be ignored. These bits are required for error detection and will create the same overhead (in terms of bits per second) independent of the configuration. The numbers to be presented assume a 4-way interleaved M2 corresponding to a peak rate of 128 Mbps.

Consider first the conventionally structured AU and I/O without an M1. From Table 7-3, we notice that a 32 bit wide bus will not satisfy a single time multiplex configuration for the average rate. Also, the dual time multiplex bus will just about carry the average required load. However, it will be busy most of the time. As a matter of fact, both parts of the time multiplexed bus will be busy 57 percent of the time. This will tend to reduce overall performance.

With the dedicated bus structure, the bus contention is not a factor. However, heavy bus traffic will cause M2 to be busy for a significant percentage of time. If M2 can deliver information at a 128 Mbps rate, each AU will require M2 $(86/128) = 67$ percent of the time. Memory conflict is now significant.



INCREMENTAL
DELAY

TOTAL DELAY = 150 NS BETWEEN Clocked FLIP-FLOPS

Figure 7-9. Internal Bus Delay Factors

It is clear that a HOLM structure minimizes bus and memory conflicts compared to conventional structure.

Consider a single time multiplexed bus with HOLM structured AU's. If one AU desires bus access, what is the probability that the bus is busy? The other AU and the I/O unit impose an average traffic of 56 Mbps. The probability that the bus is busy, given an AU is making a request, is $56/128 = 44.6$ percent. To compensate for the bus being busy and the processor having to wait, the desired system peak performance must be increased by 44.6 percent to maintain the same overall throughput. The probability that the dual time-multiplexed bus is busy is 4.1 percent.

The dedicated bus only faces the problem of memory contention. Assuming two M2 complexes, with each module possessing a 4-way interleaved structure so as to achieve 128 Mbps, and assuming that information is randomly stored in both M2 complexes, what is the probability of a memory module being busy given a request by an AU?

The bit rate imposed by one AU and the I/O is 56 Mbps. On the average, half or about 28 Mbps, comes out of each M2 at a peak rate of 128 Mbps. Therefore, an M2 complex is busy $28/128 = 22$ percent of the time. The memory conflict problems are the same for either bus structure.

By more intelligently storing information in M2, the busy probability can be reduced. The 22-percent represents a worst-case figure. With a 4-way interleaved memory, four simultaneous communications can ensue. This means that even with a 3-port memory system the worst delay encountered is two memory cycles.

On the basis of the above discussion, the dedicated bus with HOLM AU's and I/O structures is the most desirable.

7.2.3.2.5 Hardware Considerations. Table 7-4 indicates the number of drivers and receivers needed for each configuration. The three configurations must be structured so as to yield the same performance. If the dual time-multiplexed bus is W bits wide, then an equivalent single time-multiplexed bus must be $2W$ bits wide to achieve the same data rate with the same technology (bits per second per wire). For a 2-AU, 1-I/O, 2-M2 configuration the dedicated bus width can be less than the dual time-multiplexed bus. However, for the purpose of discussion, we will assume they are equal and still show that the dedicated bus structure requires less drivers and receivers.

We notice that for two AU's, one I/O and two M2's that the dedicated bus requires 10-percent less drivers. For two AU's, one I/O and three M2's all the configurations use the same number of drivers and receivers. For four or more M2 complexes the dedicated bus loses out.

7.2.3.3 Internal Bus Recommendations

As a result of the preceding analyses, including an examination of the operational multiprocessor requirements, a recommendation for the dedicated bus with multiport memories is made. Table 7-5 presents an evaluation of the various tradeoff factors.

Table 7-4. Comparison of Number of Drivers and Receivers

	Configuration		
	1	2	3
Bus width	2W	W	W
Number of AU's = A			
Number of I/O units = I			
Number of M2 modules = M			
Number of ports per AU	1	2	1
per I/O	1	2	1
per M2	1	2	A+I
Number of driver/receiver pairs for			
each AU	2W	2W	W
each I/O	2W	2W	W
each M2	2W	2W	W(A+I)
Total number of driver/receiver pairs	2W(A+I+M)	2W(A+I+M)	AW+IW+MW(A+I)
A = 2, I/O = 1, M = 2	10W	10W	9W
A = 2, I/O = 1, M = 3	12W	12W	12W
A = 3, I/O = 1, M = 2	12W	12W	12W
A = 2, I/O = 1, M = 2	12W	12W	12W
A = 2, I/O = 1, M = 4	14W	14W	15W

Table 7-5. Internal Bus Evaluation Summary

1 - first choice 2 - second choice 3 - third choice				
Factors	Note *	Configuration **		
		1	2	3
Performance	1	1	1	1
Expandability	2	1	1	1
Timing problems	3	3	2	1
Bus contention	4	3	2	1
Bus control	5	3	2	1
Bus-M2 interface control	6	1	2	3
Bus-P interface control	7	1	2	1
Development risk	8	2	3	1
Bus hardware	9	2	2	1
* Note explanation on following page **Configuration 1 - Single time-multiplexed bus Configuration 2 - Double time-multiplexed bus Configuration 3 - Dedicated buses with multiport M2				

Notes of Explanation for Table 7-5

1. All the configurations were normalized to achieve equivalent performance.
2. Expandability is not a major consideration since it is provided for in the initial sizing of the central processor.
3. The dedicated bus (Configuration 3) possesses the least timing problems, since the bit rate on each leg of the bus is less than for Configurations 1 or 2 (See Table 7-3).
4. There is no bus contention in Configuration 3. Configuration 1 has the most (paragraph 7.2.3.2.4).
5. Control of the time-shared bus requires logic to poll the communicating elements and resolve bus conflict. A separate bus control element must be designed. The dual time-multiplexed bus must also control which bus is used by which communicating pair. The dedicated bus control is distributed within the M2 complex. Bus conflict is not a problem.
6. The 3-port M2 interface used in Configuration 3 is the most complex. It must provide buffer storage for three M2 requests and a scanner multiplexer element. It must also resolve memory conflict. The single time-shared bus has only one port. Conflict does not exist at the M2 interface since it is resolved by the bus control logic. Configuration 2 must provide logic to resolve simultaneous access by two buses into a single M2.
7. The bus AU interface, as well as the bus I/O interface, is the same for Configurations 1 and 3. Both have single ports. Configuration 2 may communicate on one of two buses and provides a slightly more difficult situation. Most of this problem is resolved by the bus control logic.
8. The dedicated bus has been used in existing aerospace multiprocessors; for example, the CDC Alpha. Single time multiplexed buses have been used mostly for I/O channels. Dual time multiplexed buses have not, to the author's knowledge, been applied in computer systems, although they are commonplace in telephone switching networks.
9. For the two AU, one I/O, two M2 configuration, Configuration 3 needs 10-percent less interface circuits (see Figure 7-9). For larger configurations, the balance favors the time-multiplexed bus. However, we are considering the configuration implied by the performance and capacity requirements.

1. The bus structures were normalized to provide the same performance.
2. Section 2.0 indicates that the difference between the initial design requirements and the maximum design requirements is only 50 percent. This limited expansion capability can be achieved internal to each of the communicating elements. Therefore, expandability is not a factor in evaluating the various internal bus structures.
3. The switching hardware for the internal bus is assumed to be packaged within the M2 complexes.
4. In evaluating the internal bus structure, a limited multi-processor consisting of two AU's, one I/O, and two M2 complexes was used as a basis.

7.2.4 Fault-Tolerant Central Processor

The basic fault tolerance requirements were discussed in Section 6.0 of this report. Subsequent to the study reported in that section, the criteria were reviewed at a DPA design review meeting. Table 7-6 presents the modifications and clarifications of the DPA failure and error tolerance criteria. The most significant change in requirements is that after failure detection, computational processing may be suspended for up to one minute so that failures may be isolated, the system reconfigured and the restart procedure initiated.

It is assumed, in the configuration to be described, that this suspension of computation and control for one minute is satisfactory. However, we have tried not to lean too heavily upon this requirement. All the functions performed during reconfiguration should not require more than 100 ms and at most one second of computation to commence a retry operation.

The one-minute interruption is still considered to be within specification as far as the definition of "Fail Operational" is concerned.

7.2.4.1 Philosophy

The following points of view have been formulated after a reevaluation of the requirements.

1. All critical programs and variable data should be permanently resident in M2.
2. Backup for critical programs will reside in M3. The details of this mechanization will be discussed shortly.
3. All noncritical programs and variable data will reside permanently in M3 and be brought into M2 using the preplanned overlay techniques discussed in Section 7.2.2.2.

Table 7-6. DPA Failure and Error Tolerance Criteria

Failure and Errors	Detection		Reconfiguration			Computational Capability After Reconfiguration
	Coverage	Reaction Time	Coverage	Reaction Time	Action	
First	100%	*Immediate	100%	1 Minute	Automatic Reconfiguration	**Critical station function and all experiment operations
Second	100%	Immediate	100%	1 Minute	Automatic or Manual Reconfiguration	Critical station functions
Third	100%	Immediate	100%	1 Minute ***	Automatic or Manual Reconfiguration	None - Emergency mode

* Immediate - Before erroneous output signal is transmitted to a subsystem and before reaching a state where reconfiguration is impossible.

** First failure shall not interrupt experiment computation

*** Emergency indication shall be sent to all subsystems and the DPA turned off

4. Application programs will be structured so that they are restartable with a minimum of initialization. Only critical tasks need be restartable. Each task must be analyzed in detail so it can be segmented into restartable jobs. Although this software effort can be considerable, it can only be circumvented by designing a completely triple or quadruply redundant system with triple voting elements. This includes AU, I/O and M2 as well as the internal bus and data bus.
5. The responsibility for enabling a function to fail safe shall not be placed upon the software within the central multiprocessor. The responsibility shall reside with the particular subsystems external to the data bus. The only imposition upon the central multiprocessor is to communicate to the outside world that a fail safe condition exists (i.e., following the second failure within the central multiprocessor). The external subsystems must be designed to terminate operations smoothly, or to take other appropriate action to prevent the propagation of the error conditions. Control units must be designed to prevent lockups, and to not accept erroneous sensor or command data.
6. Should a failure occur within the central processor, it will be reconfigured to handle only critical functions. This may include deactivating operational hardware. Full operational status is not reestablished until detailed fault isolation routines are executed and the faulty IFRU is replaced.
7. The one-minute reconfiguration time implies that traffic on the data bus may be suspended for up to one minute during the reconfiguration cycle.
8. Finally, a second failure is assumed to not occur during the reconfiguration cycle.

7.2.4.2 Fault-Tolerant System

Figure 7-10 depicts the block diagram of the fault tolerant multiprocessor. Each element will be discussed separately, and then the interaction of the elements to provide fault tolerance will be described.

7.2.4.2.1 Internal Redundancy. The fault-tolerant multiprocessor consists of two dual redundant AU complexes. Two complexes are required to meet the performance required. Internal to each complex are two AU's operating in a completely phase-synchronized manner. The outputs of the two AU's are compared in a comparator C. Only with two AU's and a comparator can the goal of 100-percent failure detection be achieved. This failure detection requirement is necessary for both critical and noncritical tasks. When the C element detects an error within the AU complex the following actions ensure:

1. All four AU's will be notified.

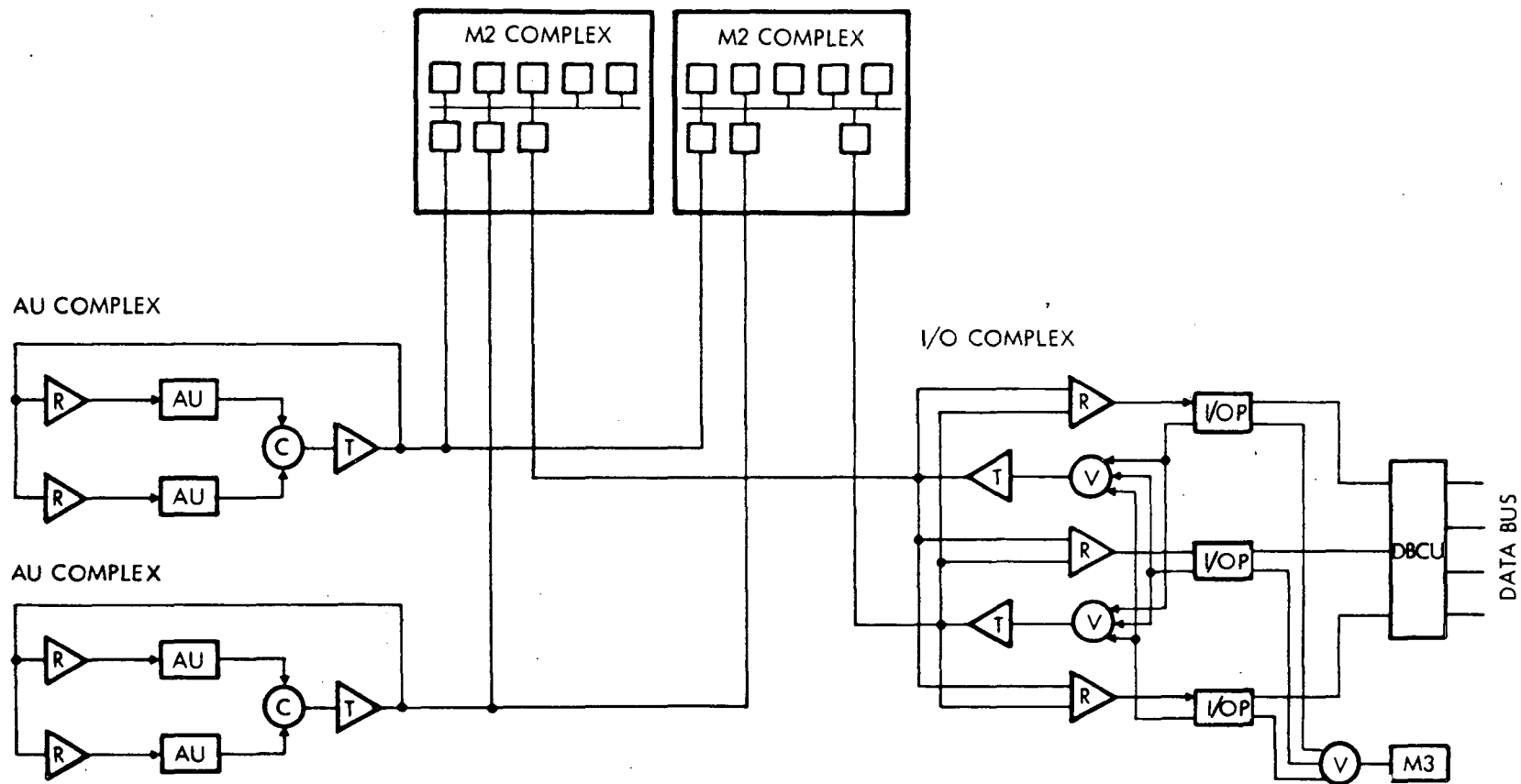


Figure 7-10. Fault Tolerant Multiprocessor



2. The AU complex without the error will notify the executive (resident in M2) so the failed AU complex will be made a non-available resource.
3. The system will revert to the fail operational mode, with only the good AU complex executing critical tasks.
4. The good AU complex will issue a signal to reset all registers and flip flops in the other AU complex and initiate its hard-wired start-up sequence.
5. After the failed AU complex is restarted it will be made an available resource again and assigned scheduled tasks.
6. If no more error indications occur, then it is assumed that the error was a transient and the noncritical tasks are resumed. If another error indication is obtained a hardware failure is suspected. Steps 1 through 5 should be repeated at least twice before the firm determination of a hardware failure is made.

7.2.4.2.2 M2 Fault Tolerance. The central multiprocessor will consist of two M2' complexes, each with half of the required capacity. The concept of an M2 presented earlier has been modified by absorbing the functions of M2' into the memory complex itself. The storage of the error checking bits is still contained in a physically independent memory module. Both M2 complexes along with the AU's and I/O's are used to meet the fault tolerant requirements.

The major functions of the M2 complexes are as follows:

1. Each M2 complex consists of a four-way interleaved memory.
2. The fault-tolerant structure possesses a fifth internal module which is used for storage of error checking information.
3. Each internal memory module can be expanded up to 16K words.
4. The AU or I/O interfaces provide an error detection function for both memory and internal bus. They also serve as a command and address decoder as well as a data buffer for the five memory modules.

The error detection features incorporated into the M2 complex consist of:

1. Every word written into a memory module will contain a parity bit generated by the AU or I/O. This makes every memory word 33 bits long.
2. Every access to the M2 complex will consist of a 4-word block, with each word stored in a separate memory module.



3. Program instructions and data are contained in memory modules one through four. The fifth module contains a special vertical check word. This word corresponds to the "exclusive OR" of the vertical parity word, described earlier, and the physical block address.
4. The memory interfaces will be capable of responding to a number of different commands.
 - a. Read and check. This causes the five memory modules to be read and sent to the requesting unit: data parity bits, check words and all. The R element of the AU or I/O will verify the parity bits and check word. This also checks the internal bus transmission. Although the transmission of the fifth check word increases the bus traffic by 20 percent, the ratio of average to peak transfer rate will be able to absorb this increase.
 - b. Write and verify. This command transmits information from the AU or I/O with parity bits and check word generated by the AU's already attached. Upon receiving the five-word block the interface logic at M2 will check the word parity bits. The vertical parity check word will be calculated and the block address extracted. This will be compared with the storage address sent in the initial address and command word.

After the block is stored in the M2 modules it will be read out again and the read process verification sequence described in the read and check function executed.

The write and verify command increases the bus traffic by 240-percent for write operations. However, most of the M2 accesses are not expected to be write operations. The write and verify command is only required for the final results of computations. Temporary variables are contained in the evaluation stack resident in M1. If every instruction were a store operation, only 50 percent of the memory accesses would be write operations. If every instruction pair were a load followed by a store, then only 25 percent of the operations would be writes. Considering a more realistic dynamic instruction stream and realizing that temporary variables need not be written into M2, it is estimated that no more than 15 percent of the memory accesses are write operations. Therefore, the bus traffic would only be increased by 36-percent on the average.

Table 7-3 indicated a 40 Mbps requirement for the average bit rate on the internal bus. However, the design goal was chosen to be 128 Mbps to handle peak requirements. Both read and write verify commands do not make the average bus rate exceed the peak bus rate. The average is increased by 21.2 Mbps for a total of 61.2 Mbps.

These two commands will satisfy failure detection criteria within M2.

Both of the M2 complexes must work together to meet the overall fault tolerant requirements. Each M2 is divided into a number of distinct areas. One area contains variable information associated with critical tasks. Certain tables associated with the executive function fall into this area. This area is denoted by CD. The second area is associated with resident critical instructions. This is denoted by CI. The final area contains images of non-critical instructions and data, a copy of which is permanently in M3. This area is denoted by NC. For details see Figure 7-11.

The CD area in both M2 complexes is always written into redundantly. The information contained in CD is necessary to restart critical functions in the "fail operation" mode. The two CI areas contain different instructions in each M2. If this area of memory fails, the one-minute reconfiguration interval is more than sufficient to read copies of the programs from M3. If the NC area of memory fails recovery is not necessary. Only the detection of the failure and the appropriate fail safe sequencing is required.

7.2.4.2.3 Internal Bus. With redundant dedicated bus structures, a bus segment could fail and still the AU or I/O would keep working. Further reflection upon this problem leads to the conclusions that if one dedicated bus segment failed the AU that drives that bus could be reconfigured out of the system until the bus was repaired. The other AU complex with its bus could handle the critical functions until the system was repaired.

For the I/O functions, two buses are shown originating from the I/O complex. Each bus intersects a different M2 complex. The failure of one I/O bus forces one M2 complex to be reconfigured out of the system; however, the other M2 complex has enough capacity to execute all critical I/O tasks.

7.2.4.2.4 Input/Output Complex. It has been determined that one I/O unit is capable of handling all data bus traffic, M3 traffic and the necessary limited instruction execution.

Figure 7-10 shows a triple redundant complex with voters.

The data bus control unit (DBCUC) is shown as a single interface between the triple redundant I/O and the quad redundant data bus. To meet the fault tolerant criteria it would probably have to be triple or quad redundant. The design of the DBCUC is considered to be external to the central multiprocessor.

The mass memory is shown interfacing to the I/O complex.

7.2.4.2.5 M3 Considerations. M3 need not possess any reconfiguration capability since no critical information is resident in M3. A 100-percent error detection capability must still exist within M3 to provide a fail safe capability for the overlayed noncritical functions. When M3 fails it is configured to be off-line. Critical functions can still be executed since they are resident in M2.

The error detection logic for the pages stored in M3 can be similar to that proposed for M2. It should contain both horizontal and vertical parity as well as physical address information within each stored block. Echo checking for write operations is suggested.

7.2.4.2.6 Restart Software. The ability to restart a process after it has been abruptly stopped due to failure detection has been assumed in the previous sections. This is well known to be a difficult and fallible task. A number of suggestions are made to make it easier.

1. All restartable program elements must be constructed from reentrant code. That is, code that does not modify the instruction stream by execution of the program.
2. All the results from the execution of a routine should be accumulated and written into the CD area of memory at the end of the routine.
3. The redundant CD areas in the two M2 complexes should be written sequentially in time so as to prevent the failure in an AU complex from causing both copies to be written incorrectly.
4. All restartable programs should be structured to operate upon one set of input parameters presented in one time. These input parameters are stored redundantly in CD. They are not destroyed until after the final results have been redundantly written into CD and verified.
5. Only tasks scheduled by the executive are restartable entities.
6. Tasks which are at lower levels of priority or are called by restartable tasks, will be completely reexecuted by execution of the main task. All parameters generated by these lower level utility tasks are temporary and will be reevaluated.
7. The interaction between reexecutable tasks should be minimized and only handled at the highest executive level.
8. Critical program instructions and variables may be conveniently identified when the HOL program is written. The critical variables may be identified in the initial declare statement. Variables not declared are considered temporary. They can be recalculated and are not critical. The program name may contain information as to its criticality. The compiler or link loader can utilize this information when M2 space is allocated.

7.2.5 Summary and Recommendations

The final conclusions and recommendations concerning the elements internal to the operational multiprocessor are presented in this section.

7.2.5.1 Architectural Considerations

The internal structure of the multiprocessor will consist of:

1. Two independent AU complexes
2. Two independent 3-port M2 complexes
3. One I/O complex
4. Internal buses dedicated to the two AU's and I/O unit
5. An M3 storage device interfaced to the I/O complex
6. Expansion will be internal to each communicating element

7.2.5.2 M1 and AU Considerations

The local storage, M1, and the AU form an integral structure and must be discussed as one unit. The following recommendations are made.

1. The internal structure of the AU and the utilization of M1 shall be such as to cause an efficient execution of higher order language statements stored in M2.
2. The functions incorporated into M1 include:
 - a. Evaluation stack
 - b. Descriptor stack
 - c. Temporary storage of variables
 - d. Buffer for the polish string
3. Associated with the descriptor stack will be a small associative memory of about 64 words.
4. The total size of M1 should be less than 400 - 32 bit wide words.
5. The addressing of M1 shall be down to the byte (8 bit) level.
6. Speed requirements shall be between 200 ns and 500 ns cycle times.
7. Error detection is accomplished by utilizing two AU's and a comparator in each AU complex.
8. The recommended M1 technology is CMOS, with bipolar and plated wire as alternative.

7.2.5.3 Input/Output Considerations

The following recommendations are made concerning the I/O complex.

1. The I/O processor is a simpler structure than the AU.
2. It possesses local storage sufficient for buffering data transfers.
3. M3 is controlled through an independent I/O port.
4. The I/O unit performs the following functions:
 - a. Instruction execution for data bus control
 - b. Instruction execution for M3 control
 - c. Buffering of data bus commands and data
 - d. Partial buffering of M3 data transfers
 - e. AU interrupt control and priority
5. Fault tolerance will be achieved through triple redundancy with voters.
6. The DBCU will contain a hardwired fail safe sequence for the second failure.

7.2.5.4 M2 Considerations

The following conclusions have been reached concerning M2.

1. M2 performs the following functions:
 - a. Stores all critical functions
 - b. Redundantly stores critical variables
 - c. Provides an overlay area for M3 storage of noncritical functions
2. The M2 memory complex possesses the following features:
 - a. Five-way interleaved memory modules
 - b. Independent AU and I/O interfaces through 3 ports
 - c. Block-oriented organization with 4 words plus one check word per block
 - d. Each word is 32 bits plus one parity bit
 - e. Each M2 memory module is expandable from an initial 8K words
 - f. Upward expansion to 16K for each memory module is sufficient
3. M2 cycle time is approximately 1 microsecond for a memory module. Overall throughput to the internal bus is 160 Mbps with five-way interleaved memory.

4. There is a 1 microsecond latency for each read operation.
5. The maximum delay due to memory conflict between the two AU's and the I/O is 2 microseconds.
6. The technology suggested is plated wire.
7. The maximum capacity of each M2 complex is less than 80K words including parity words.
8. Error detection is accomplished by:
 - a. Word parity bits
 - b. Vertical parity words exclusive ORed with the physical block address
 - c. Echo checks of all write operations

7.2.5.5 M2-M3 Transfer Control

The following decisions have been reached concerning the relationship between M2 and M3.

1. There will not be a virtual memory system.
2. Preplanned overlays will be used to map M3 into the 16K overlay area in M2.
3. The overlays will be determined within each mission phase by time-lining non-overlapping functions.

7.2.5.6 M3 Considerations

1. The functions of M3 are
 - a. To store all noncritical instructions and data
 - b. To store redundant copies of critical programs which are resident in M2
2. Nothing stored in M3 is critical and therefore M3 recovery after failure detection is not required.
3. One-hundred-percent failure detection is required. This is accomplished by:
 - a. Word parity bits
 - b. Vertical parity
 - c. Imbedding physical address information in each block
4. Two technologies can be anticipated. A drum will provide a low-risk option and implies a 5 millisecond latency time.

A large plated wire memory is a higher risk option. No mass plated wire memories have been built. Although the latency time for plated wire is less than for a drum, it is not a critical factor in the CP configuration proposed. Either technology will suffice as far as requirements are concerned.

7.2.5.7 Internal Bus Considerations

The following conclusions have been reached concerning the nature of the internal bus.

1. The bus structure consists of a dedicated bus link for each AU and I/O complex, requiring each M2 complex to possess three ports.
2. The bus is 32 data bits wide plus one parity bit, with an unspecified number of control lines.
3. The maximum bit rate on each wire shall be 5 Mbps.
4. The peak transfer rate on any bus link shall be 160 Mbps.
5. There is no bus contention with the dedicated bus.
6. Error detection in the internal bus shall be by means of word parity bits.

PRECEDING PAGE BLANK NOT FILMED

7.3 REFERENCES

1. Sugimoto, M., "PL/I Reducer and Direct Processor," Proc. 24th National Conference, ACM, 1969.
2. Kerner, H. and Gellman, L., "Memory Reduction Through High Level Language Hardware," AIAA Journal, Dec. 1970, pp. 2258-2264.
3. Graham, R. M., "Use of Higher Level Languages for Systems Programming," Technical Memorandum 13, Project MAC, September 1970, AD 711 965.
4. Corbato, F. J., "Sensitive Issues in the Design of Multi-Use Systems," MAC-M-383, Project MAC, December 12, 1968.
5. Corbato, F. J., "PL/I as a Tool for System Programming," Datamation, May 1969.
6. Carey, L. and Sturm, A. A., "Space Software: At the Crossroads," Space/Aeronautics, Dec. 1968, pp. 62-69.
7. Bostrom, F. D., "Higher-Order Language Study for Avionics Programming," IBM Technical Report, AFAL-TR-154, June 1971.
8. Rice, R., Smith, W. R., "SYMBOL - A Major Departure from Classic Software Dominated Von Neumann Computing Systems," Proc. SJCC 1971, pp. 575-587.
9. Keeler, F. S., et al., "Computer Architecture Study," AF/SAMSO Deport TR-240, October 1970.
10. Summer, F. H., "Operand Accessing in the MU5 Computer," Proc. 1971 IEEE Int. Computer Society Conf., Sept. 1971, pp. 119-120.
11. Raytheon Co., "Computer Display Channel Design Plan, Vol. 1," October 1968. FA67NA-27.

8.0 MEMORY STUDIES

8.1 MASS MEMORY TECHNOLOGY

8.1.1 Introduction

The object of this particular task was the development of quantitative data on memory techniques that are capable of being developed into a manufacturable process for the 1975 time period. The emphasis of the study was a review of candidate memory technologies now in existence or in development. Extrapolation of the current state of those technologies is made to estimate their expected status in 1975. Where possible, estimates of physical characteristics, cost parameters and performance characteristics of each technology are indicated.

8.1.2 Selection of Applicable Technologies

In this section, we have chosen two ranges of memory capacity, 10^4 through 10^6 bits ADT effort. These two ranges have been chosen since they encompass the range of memory capacities for operating and bulk storage that are being considered as part of the task to establish the baseline memory hierarchy. The M3-1 category is generally categorized as random access, with access and/or cycle times between 0.1 and 1.0 second. M3-2 may be block oriented, with access/latency times up to several milliseconds, but with high read rates of up to several million bits per second. For each of these ranges, we have selected candidate technologies which are expected to be available in 1975, and then discuss and compare them in terms of their applicability to memories of the specified size. A firm recommendation for a particular memory type is not made, however.

8.1.2.1 M3-1 Memory, 10^4 through 10^6 Bits

The list of technologies to be considered for this category is as follows:

Ferrite core, 2-D, 2-1/2-D, or 3-D

Plated wire

Post and film

Coupled film

Semiconductor

Bipolar

Static MOS

Dynamic MOS

CMOS

MNOS

For the core memory systems, any of the three organizations listed could be used for capacities up to 10^5 bits, but above this, the 2-D core arrangement becomes very expensive relative to the other organizations, and its estimated failure rate also rises rapidly. At 10^6 bits, the estimated cost of a 2-D memory would exceed \$400K, and the MTBF of the memory would be less than 1000 hours. While it is true that 2-D core can provide a faster access and cycle time than the other organizations, it is not likely that such fast cycle times will be required for this memory. And if they are, another technology, plated wire, can supply fast cycle times at a lower cost than 2-D organization. All three of the core memory technologies have destructive readout operation, which may prove to be a disadvantage in this application.

Plated wire memory overcomes the destructive readout disadvantage of ferrite cores, but is more expensive and heavier than 2-1/2-D or 3-D core. Of the existing technologies, if either NDRO operation or fast memory access and cycle times are of overriding importance, plated wire is the obvious choice.

Of the planar film memories, both post and film and coupled film appear attractive for certain characteristics. The post and film memory promises an NDRO memory of reasonable size at a cost considerably less than that of plated wire. This technology is almost ready for use. Prototype memories have been fabricated on an operating pilot production line. The coupled film memory is farther from realization.. The promise here is the very high density that should be possible using this technique. The prospect of fitting a 10^6 bit RAM memory into less than a tenth of a cubic foot is too attractive to dismiss this technique at this time. The technology is not far enough advanced, however, to allow price estimates to be made. The major hurdle for either of these technologies will be their reliability. Neither of them have been in use for long enough to allow any evaluation to be made. Until both acquire considerable mileage, they cannot definitely be considered for the space station design.

All of the semiconductor techniques discussed appear to have some applicability to this memory size except the charge coupled devices. These are omitted primarily because the minimum practical block size of 10^3 to 10^6 bits is larger than would be required at this memory level. With the exception of the MNOS device, each of the semiconductor technologies discussed here can be fabricated into either RAM or shift register memories. The comments which follow apply equally to both.

Bipolar semiconductors can achieve a fast memory, but the cost and power dissipation will be excessive for memories of over about 10^5 bits. At 10^6 bits, the power consumption would be more than one kw and the cost would approach \$200,000. The static MOS memory will be somewhat lower in power and cost, but the speed will also be less. The dynamic MOS memory has some interesting possibilities in terms of speed and density, but its dependence on the clock controlling the refresh circuitry is a disadvantage.

CMOS may turn out to be a best choice for this memory in terms of low power and high speed if the technology develops as expected in the next few years, and if the problems of radiation sensitivity and volatility can be resolved.

The MNOS memory is a non-volatile semiconductor memory which also offers a potential for high density. Again, if the questions of radiation sensitivity can be resolved, and the technology develops at a reasonable rate, this memory could be quite attractive.

8.1.2.2 M3-2 Memory, 10^6 to 10^8 Bits

For memories in this capacity range, the problem of volatility becomes more important, since the consequence of losing 10^7 or 10^8 bits due to a power transient can be severe. While recovery by reloading from an archival store can sometimes be mechanized, the time required to transfer this number of bits can be several seconds or tens of seconds, which could be unacceptable. We are, therefore, adding a non-volatility requirement to the specifications for this memory. This rules out the semiconductor memories except for MNOS. Most of them would be ruled out on power consumption or volume considerations anyway.

The candidates for this memory are as follows:

Core, 2-1/2-D and 3-D

Plated wire

Magnetic "bubble"

MNOS

Soniscan

Wire ferro-acoustic

Beam memory

Drum

Disk

Tape

The MNOS memory is a semiconductor RAM memory which combines a high density potential with non-volatility. The same reservations apply to M3-2 as M3-1 with regard to the use of MNOS in M3-2. If the problems of radiation sensitivity and the state of development of the technology can be overcome, the use of MNOS would probably still be limited to the lower end of the M3-2 capacity range due to the penalties of cost, volume, and power consumption.

The 2-1/2-D and 3-D core memories are basically RAM memories, but they could be applied to storage systems having the capacity of an M3-2. The physical limitations of the core stack constrains these technologies to the lower end of the specified range. At some point between 4×10^6 and 3×10^7 bits, the size of the core stack becomes unwieldy, and it will be desirable to break the memory up into several smaller memories. The cost is also rather high, since a 3×10^7 bit core memory is estimated to cost in the range of \$106 using either organization. A memory of this same size for 32 bit words requires a core stack that is about 1000 bits square, which is as large as can comfortably be estimated for aerospace use at this time.



Plated wire runs into the same problems as the core systems discussed in the previous paragraph, and the cost of the memory is even higher. A 10^6 bit memory might be practical but much over 10^7 will be too expensive and large. A 10^7 bit memory will cost about $\$10^6$ and occupy at best approximately three cubic feet. The advantage of plated wire over core is again the NDRO readout, the faster operating cycles and reduced power consumption. For over about 10^7 bits, neither plated wire nor core is practical for a single memory module.

The magnetic bubble memory is a new device currently in development. It has a number of characteristics that makes it attractive for this application. In volume it is projected to be much less than either core or plated wire, and through the use of a moderate number of parallel loops its total data rate can be made to match that of large core memories. The bubble memory is essentially a serial device, and the length of the data blocks in the memory significantly affect cost. There are indications that this type of memory may be economical only when the serial length of the data blocks is quite large, such as 10^4 or 10^5 bits. Since at the present time, the projected shifting rate for the memory is of the order of a megacycle, this implies a block time of 10 to 100 milliseconds to read a block, if the memory is to be economical. There is currently considerable commercial interest in this technology, with the Bell Telephone Laboratories actively developing it for use in electronic telephone switching systems. This is a definite advantage, since much of the advanced research expense will be commercially underwritten. The use of the magnetic bubble technology appears to be uneconomical for memories of the order of 10^6 bits because of the overhead expense associated with the magnet structure, etc. The range of applicability of this memory appears to start at about 10^7 bits, and continue upward from there to at least 10^9 bits.

The two BORAMs (Block Oriented Random Access Memories), the ferro-acoustic wire memory and the Soniscan memory, are complements of the bubble memory in that they are particularly adapted to block sizes of 10^4 bits and less. In addition, the serial data rates of these two memories are about an order of magnitude higher than the magnetic bubble. The volume of the Soniscan memory is an order of magnitude greater than the volume of a similar bubble memory. This consideration may limit the use of Soniscan memories to capacities of 10^8 bits or less. At 10^8 bits, the memory would have a volume of nearly 10 cubic feet. The ferro-acoustic wire technology is not well enough developed to give a good estimate of volume, but it appears to be more dense than the Soniscan and may be similar to the density of the bubble memory, or perhaps even smaller. These two memory technologies do not appear to be restricted to the very long serial lengths of the bubble memory, and therefore, the choice between these technologies may turn on the block size requirement and serial readout rate.

The beam memory technology has not proceeded far enough yet to allow a really accurate estimate of its adaptability to the aerospace environment. Particular problems lie in the effects of the environment, such as shock and vibration, on the alignment accuracies required within the memory. If they are adaptable, current indications are that this type of memory will be useable for block sizes of the order of 4×10^4 bits and less, so that it will be competitive with the Soniscan and ferro-acoustic wire rather than the bubble memory in this respect. For total capacity this type of memory does not appear to become efficient until the capacity of the memory exceeds 5×10^7 bits. The useable range of this technology is up to at least 10^{10} bits.



A major consideration with this technology is the current state of development. It appears that the only type of beam memory with much chance of being ready for use by 1975 is the mechanically deflected type. A mechanically deflected memory would suffer from the same problems that drums, disks, and tape recorders, in that it is subject to the wearout problem and will require periodic servicing. This consideration alone might dictate against the use of a beam memory in the space station. For space station applications beyond 1980, however, it will become more and more likely that the electrically deflected beam memories will be available. Thus, if the time period for this study were extended to 1977 or 1978, there could be some justification in recommending an electronically deflected beam memory.

The drum memory does not really appear to have a place in the space station computing system. In the early days of computers, the slow operating speeds of the computer meant that the few milliseconds latency time of the drum memory was not of particular consequence. With present day processors, however, these few milliseconds represent 10^3 or 10^4 instructions and this kind of a delay cannot be tolerated in operating memories.

The disk memory has a slightly longer latency time than a drum memory, but has a major advantage over it. This is that higher capacities than 10^6 or 10^7 bits are easily obtained. At least one ruggedized disk of capacity 7×10^6 bits already exists, and there appears to be no reason why memories of at least a factor of 4 or 5 larger capacity could not be built by stacking disks. The cost of such a memory, at 15 cents per bit would be one and a half orders of magnitude less than for the other already existing technologies and the weight and volume would be correspondingly less.

Although the disk is a much slower technology than the bubble and BORAM memories, it has the great advantage that it is a well understood and proven technology today. Therefore, the disk may be attractive as a backup specification for use in case the bubble or Soniscan memories do not develop as expected.

The tape recorder has the advantage of large capacity for its cost and size, and virtually unlimited capacity if the ability to change tapes is provided. This is counterbalanced by the extremely long latency time of many seconds, even when the tape is already in place. The tape recorder's place on the space station appears to be in the archival storage of experiment data, transport of data from the space station to the ground, and transport of new programs, etc., up from the ground.

8.2 ARCHIVAL MEMORY

8.2.1 Introduction and Summary

This section presents the results of a study performed to define an archival storage system for a Modular Space Station for the time period of the late 70's and early 80's. The concept presented herein reflects the procurement of an evaluation unit during the 1975-1978 time period and followed by the procurement of the flight units.

The Precision Instrument Unicon Data System was selected as the Archival Storage System to satisfy the requirements of the Space Station Program. This unit has the added flexibility that it can provide both digital data recording of the acquired data quantity and bit rates as well as store data in the analog form.

The study consisted of a review of the various documentation prepared for the archival storage system to establish an overall set of requirements. Based on this review a set of requirements were developed. A review was made of the various memory technologies to obtain potential candidate memory systems for this application.

Based on this survey many candidate memory technologies were eliminated. The most promising candidate technologies appear to be the optical devices. This selection was based on the work statement which defines an archival storage system for the defined magnetic tape equipment.

Trade offs were performed based on data supplied by vendors and discussed in the literature which resulted in the selection of the final concept.

8.2.2 Tradeoff Selection

This paragraph defines those parameters which were used as a basis for the selection of the archival storage equipment for the space station program.

When considering the factors to be used in the selection of archival storage equipment for space application, the summary considerations would include compatibility with other equipments in the system, crew interactions, interfacing repair and maintenance activity, and logistics for transportation of information and equipments between the space station and ground. Detailed factors used in the selection of the archival storage equipment are presented below though not necessarily in the order presented:

- | | |
|---------------------------------|-----------------------------|
| 1. Data quantity | 5. Physical characteristics |
| 2. Data rate | 6. Data density |
| 3. Volatility | 7. Reliability |
| 4. Development costs and status | 8. Crew interface |

Data Quantity. This factor was selected since it determines the maximum size and cost for the development of the equipment.

Data Rate. This determines the speed that the storage device must operate to handle the required data.

Volatility. Since data may be stored for a significant length time and transported between space and ground volatility must be considered to preclude loss of data.

Physical Characteristics. These are primarily size, weight, and power requirements of the unit. Their importance is obvious when considering launch vehicle weight costs and in-orbit power penalties to sustain its operation.

Data Density. Data Density was selected as a selection factor since it contributes directly to physical characteristics of the equipment and also defines the primary growth area for all devices.

Reliability. Reliability was selected to assure acquiring data as the opportunities are presented minimizing crew activity, and simplifying on-board scheduling routines.

Development Costs and Risks. Since these parameters are directly related, they are considered together.

Crew Interface. This factor is significant since it reflects into manning and operational concepts of the modular space station.

The candidates that have been selected for further study for the archival storage system are the laser beam, electron beam and holography. The trade-offs performed which led to the selection of the proposed concept are presented in Table 8-1.

Table 8-1. Archival Storage Tradeoff Table

Criteria	Laser	Electron Beam	Holographic
Data quantity	2×10^{12}	2×10^{12}	2×10^{12}
Data rate	15×10^6 bps	15×10^6 bps	15×10^6 bps
Data display	2×10^7 b/sq in.	10^7 bits/sq in.	10^8 bits/sq in.
Volatibility	non-volatile	non-volatile	non-volatile
Reliability	not known	not known	not known
Development costs	3 meg.	not available	not available
and risks	medium	higher	higher
Size	7 cu ft	{ In early development stages. Information not available. }	
Weight	235 pounds		
Power	10K w (pk)		



As previously indicated, many technologies have been eliminated. Listed below are the various candidates and rationale for their elimination in summary form:

Technology Core	Rationale for Elimination
Core	Size and weight
Thin film	Size, weight and cost
Plated wire	
Thin film	
Semiconductor	Size, weight and cost
MOS, CMOS, etc.	
Magnetic storage	Size and weight
Disks and drums	Reliability and quantity of tape with its associated weight and volume penalty
Magnetic tape	
Bubbles	Feasibility established. No operating memories available in early development to be able to effectively evaluate.

Based on results of the study and available information on the various memory technologies, the concept proposed for the space station archival memory is an optical device utilizing the laser beam. A candidate system being proposed is the Unicom Space Data System of the Precision Instrument Company. This concept was selected since it satisfies all requirements of the archival storage system.

Since ground-based systems have been delivered and are in operation, projections for the time period of interest indicate that such a system can be developed for space use.

9. DPA CONFIGURATION SELECTION

9.1 REVISIONS TO BASELINE DPA CONFIGURATION

Subsequent to the selection of a baseline DPA configuration (see Section 3) several of the influencing factors were changed as a result of the concurrent MSS Phase B definition studies. The buildup sequence shown on Figure 9-1 was selected in favor of the previously chosen sequence (power, core, etc.). The DPA failure and error tolerance criteria were redefined, see Table 7-6. The computational requirements were also redefined, see paragraph 2.2.1.

Figure 9-2 presents the configuration selected for the Data Processing Assembly. Two identical central processors are utilized - one for station operations and the other experiments and backup for station operations. The station operations Central Processor (CP) is located in the primary control module (SM-1). Supervisory control of the equipment in the Power and Core Modules is provided via a radio link during station buildup prior to SM-1 arrival. A special component (Buildup Data Processor) is located in the Core Module for interfacing with the radio link and the MSS subsystems. This component will be removed or disengaged when SM-1 arrives and supervisory control exercised by the station operations Central Processor.

The baseline configuration is further shown to consist of Remote Processing Units (RPU) performing G&C subsystem functions and failure detection. A redundant bus network connects these RPUs and Remote Acquisition and Control Units (RACUs) to the central processor. A multiprocessor organization has been selected as the most suitable for the central processor. Redundancy at the central control level is further supplied by the other central computer containing the critical operations function and experiment support software.

This second Central Processor is located in another pressure volume (SM-4) and is identical to the primary computer. The RPUs consist of uniprocessors with special input/output processing or signal processing as required to accommodate the subsystem functional requirements.

Figure 9-3 gives a block diagram of the new baseline DPA showing the number and distribution of processors and RACUs.

Safety of operation is provided through use of redundancy of equipment and location. The two control centers are located in two separate pressure volumes. Interconnection is provided with a multiple bus network. Maintenance is facilitated with an OBCO system which includes the monitoring of signals and the ability to fault isolate with either automatic or man-generated check-out programs. Other features of this DPA are commonality arising from similar components and few types; flexibility due to the bus structure, incremental buildup capability and interchangeability of components; and operational availability due to several levels of redundancy and degraded modes of operations.

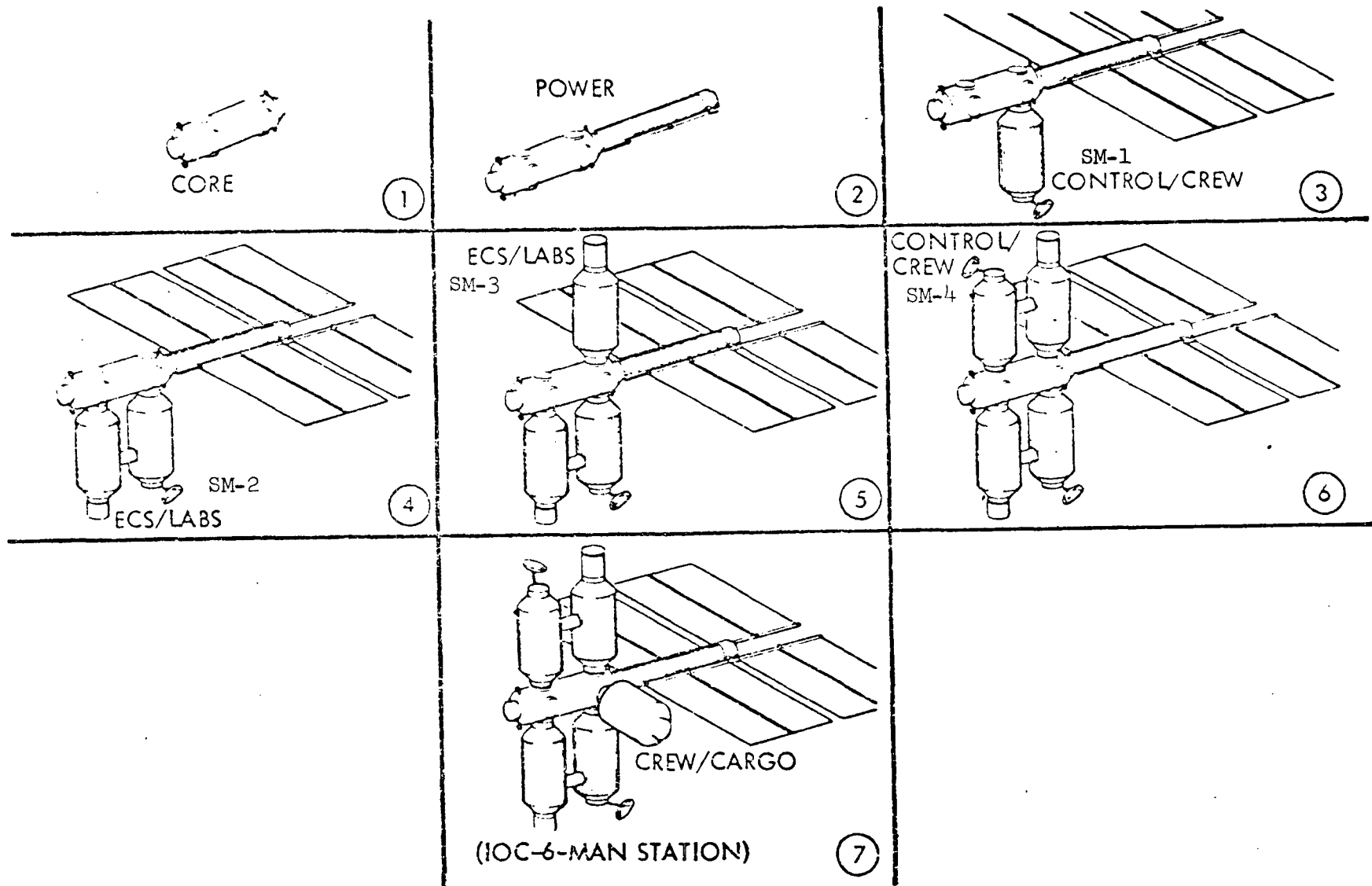


Figure 9-1. Buildup Sequence - Initial MSS

* Not part of DPA

9-3

SD 72-SA-0114-4

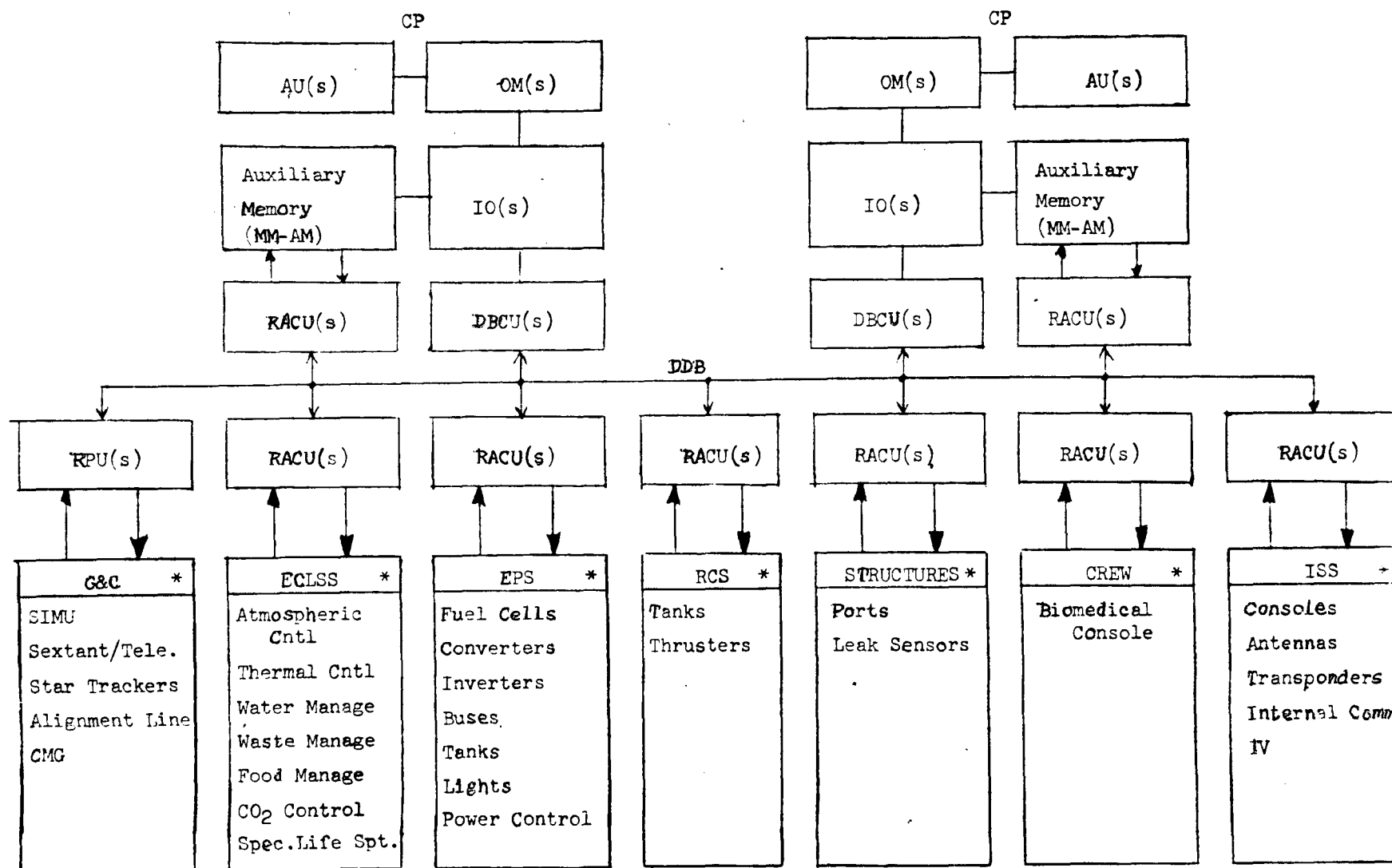


Figure 9-2. DPA General Diagram

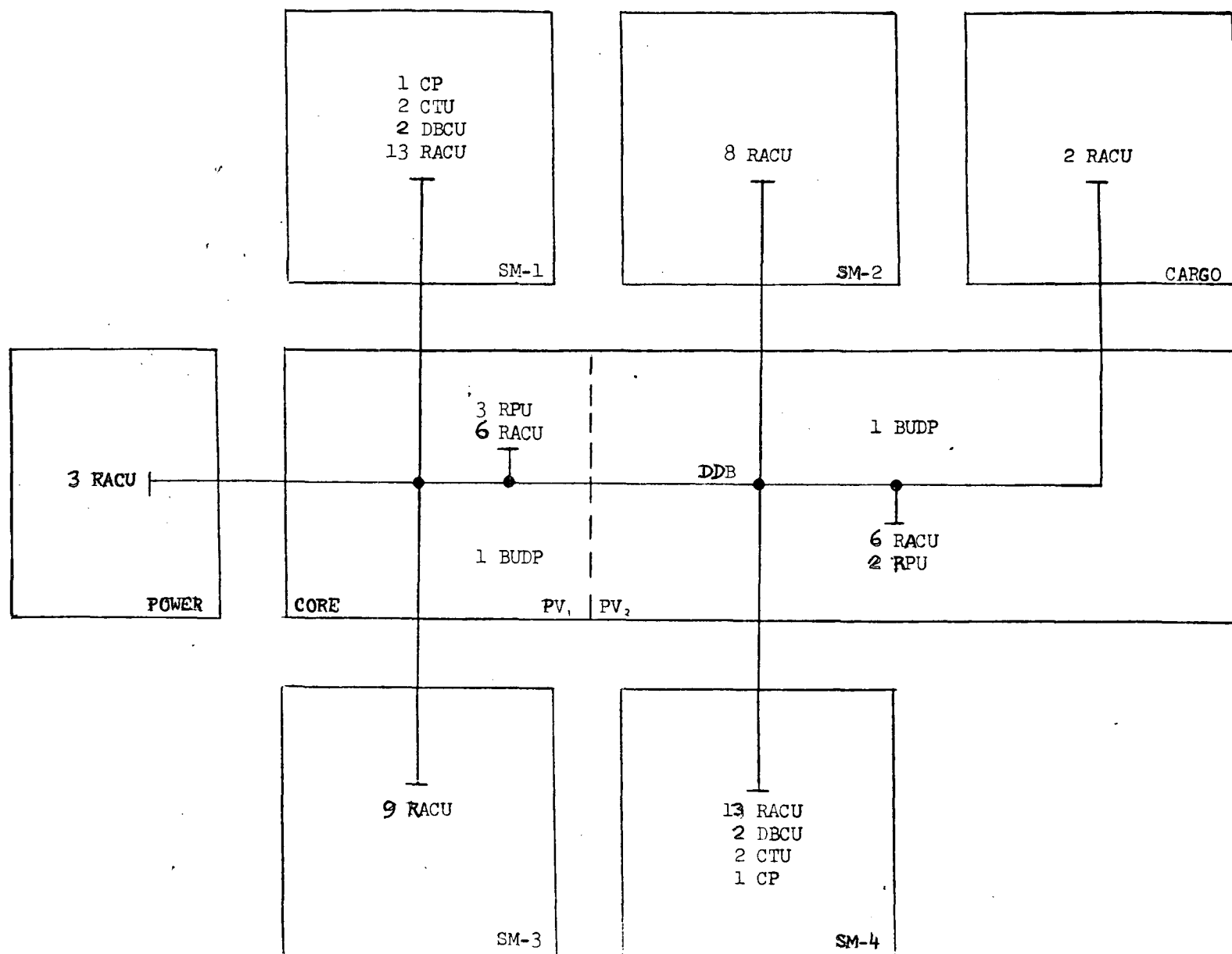


Figure 9-3. Revised DPA Diagram (Six-Man Station)



Components of the DPA are realizable with the present aerospace state-of-the-art. Future improvements in physical characteristics and cost are possible with the expected technology changes in memories (solid state and plated wire) and logical devices.

The baseline configuration also offers the system advantage of permitting subassemblies and subsystems to be operated and checked out prior to total system integration.

The central computing complex supervises the preprocessors and controls the communication with the space station and ground subsystems. It supplies the spacecraft and mission management operation and overall fault isolation plus crew interface. The central processor has access to the measurements and control points within the affected subsystems through the RACUs.

The configuration presented here is for a 6-man level. The growth to the 12-man station is accommodated by increasing the memory sizing and adding RACUs to accommodate the increased power load.

The experiment or backup central processor is made identical to the operational or primary central processor. Its normal operation would be to hold critical programs in its operating memory. Periodically data would be supplied to these programs to provide a reference point in the event of reconfiguration for a primary processor failure. The remainder of the computer is devoted to servicing the experiments. Upon reconfiguration, i.e., after two failures to the primary CP, the required operational programs (loaded from mass memory) are performed and the experiment support permitted to degrade.

Table 9-1 presents the resultant computational requirements for the Central Processors. As can be noted, a breakdown for critical and non-critical sizing is provided. Further, the sizing estimates were doubled to arrive at the estimate for the initial station. This allowance is for the uncertainties in the estimation. An allowance for growth to a final capability is shown. The initial DPA equipment would be designed for these final values. It is to be noted that, at this stage, no allowance has been included for the additional speed and memory required for error detection internally in the central processor. Further, the allocation of non-critical information to the operating or mass memory was made on the basis of iteration rate. If the iteration rate of a function was greater than 1/second, the corresponding information was assumed to be in mass memory, otherwise in the operating memories. The archival memory contains the data base and programs.

The requirements for the preprocessors which are used in the Remote Processing Units (RPU) and in the Guidance and Control Subsystem is shown in Table 9-2. Since the computational loads are well defined by previously sponsored NASA studies, since the central processor can accommodate any increased computational loads, and since spare capability is provided in order to make all preprocessors alike, the initial and final values for the memory and speed were made equal and twice the sizing estimation.

The RACU quantity and subsystem interface information is presented in Table 9-3.

Table 9-1. Central Processor Computational Requirements

	(MEMORY - 32 BIT WORDS/SPEED - EAPS)		
	SIZING EST. (X)	INITIAL (2X)	FINAL (3X)
CP - CRITICAL	36.3K/217K	72.6K/434K	108.9K/.65M
CP - NON-CRITICAL	30.7K/414K	61.4K/828K	92.1K/1.24M
CP - TOTAL	67K/631K	134.0K/1.26M	201K/1.89M
MASS MEMORY	341K	682K	1024K
ARCHIVE MEMORY-WDS	4.2M	8.4M	12.6M

Table 9-2. RPU Sizing

	MEMORY/SPEED (X)	2X(INITIAL) (FINAL)
RPU	(1)	(2)
1 (RCS)	4700/55600	9300/111,200
2 (CMG)	2600/39800	5100/79,700
3 (ORA)	2500/800	5100/1600
4 (RCS)	4700/55600	9300/111,200
5 (IRA)	4400/69,600	8800/139,300

(1) Includes 5% for Executive

(2) A single RPU Design of 10,240 words (32 Bits) and 150 KEAPS is recommended.



Table 9-3. RACU Utilization

	RACU NO	IN			OUT	
		A	E	D	E	D
C O R E	PWR 1	84	42		13	
	2	97	45		34	1
	3	34	16	1	30	1
	PV1 4	128	29	1	128	
		128	29		83	1
		128	29	1	128	
		128	19		83	1
		103	67	1	54	1
		82	86	1	47	1
		128	19	1	128	
	PV2 11	128	19		83	1
		128	19	1	128	
		128	19		83	1
		128	19		83	1
		115	62	1	52	1
		102	106	1	51	1
	SM1 16	111	54		65	1
		111	80		86	1
		128	128		128	
		128	128		128	
		85	49		47	1
		40	4		19	
		105	72		55	
		35	63	1	36	1
		66	35	1	44	
		(NOT SIZED)				
			(NS)			
			(NS)			
			(NS)			
	SM2 29	107	75		78	1
		107	75		78	1
		44	48	1	26	1
		128	63		23	
		128	16		39	
		99	54		44	
		86	54		43	
		115			22	
	SM3 37	107	75		78	1
		107	75		78	1
		21	48	1	26	1
		113	102		10	1
		128	63		23	
		128	16		39	
		99	54		44	
		86	54		43	
		115			22	
	SM4 46	111	54		65	1
		111	80		86	1
		128	128		128	
		128	128		128	
		85	49		47	1
		40	4		19	
		105	72		55	
		35	63	1	36	1
		66	35	1	44	
			(NS)			
			(NS)			
	CARGO 59	30	44		11	1
		47	16		11	
	RPU 1	24			24	
		52	16		24	1
		18	33	1	21	1
		24			24	
		27	81	1	61	1



The dual redundant bus permits the intercommunication of either of the central processors (CP) with any subsystem or element of the DPA. There are a number of ways, as will be illustrated, by which the DPA may operate. The selection of the most suitable requires further studies including those regarding detailed definition of the experiments and subsystems.

One concept of normal operation is to assign one CP to perform the station operations and use one bus in each pressure volume while the other CP does the experiments and uses the other buses. The RACUs are assumed to be capable of only communicating with one bus at a time. If a conflict occurs, the CP requesting access to a particular RACU can note whether transmission is being conducted with the other and idle until servicing can be accomplished. Alternately, the operational CP could have priority over the experiment CP and be serviced immediately.

Another concept of normal operation is to allocate time slots for particular RACUs and communicate accordingly. Since the bus rate has been chosen to meet the requirements for both experiments and operational data flow, sufficient time slots are provided.

Still other methods of operating are possible. For instance, identical data can be transmitted on two or more buses. Such bus usage permits data comparison at the receiving end. More than two may also be utilized to send identical data to all RACUs and be useful for rapid reconfiguration or as an aid to identify a fault DPA component.

Error detection and fault isolation of the DACS is performed in the central processor using software to process data derived from echo checking, command verification and validity tests. Further error detection of the DACS is obtained by use of coding on the data, format and transmission procedures and built-in self test into the DACS elements.

The central processors also use a combination of hardware and software to detect failures and to isolate faulty units. The recommended central processor concept, described in the next section, uses comparators to check all memory command, address and data. Coding of data and address verification is also used.

The G&C RPU's perform periodic self and subsystem tests. Upon error detection, the RPU can be provided programs over the data bus from archive memory for further isolation. Relocating of operational programs can also be accomplished over the bus.

9.2 CENTRAL PROCESSOR TASK ALLOCATIONS

In paragraph 5.5 where processor utilization was discussed with respect to the simulation results, the low utilization of the I/O processor was noted. It was suggested that either a lower speed be used in the I/O processor or that a reallocation of tasks between the I/O processors and the AU sets of a central processor be investigated. SDC undertook this investigation to assist in the final selection of central processor parameters by effectively allocating the estimated software workload among the central processor computing elements.

The major criteria and guidelines employed in this effort were as follows:

- a. All critical functions for the station operations CP are to be duplicated and executed in parallel in separate processing elements.
- b. Non-critical functions are to be allocated in an attempt to balance the load imposed on each "half" of the CP; that is, distributions are to be made to allocate approximately the same processing load to each of two AU-I/O combinations.¹
- c. Within each AU-I/O combination, a further distribution is to be made to evenly allocate the software load between the AU and the I/O. The ultimate goal is to attempt to use identical processor types for AU and I/O operations, thereby standardizing the types of the many processing elements used throughout the station.

Detailed breakdowns of the subsystem software loads were then made to further identify discrete portions of these loads and their expected operating times. For all subsystems the workload portions were designated as to their applicability for AU processing (i.e., relatively long-running computer-bound operations), I/O processing (i.e., loads which are comparatively short and interface frequently with subsystems through the external data bus), and processing which could be performed equally well either on AU or I/O.

Processing segments were then allocated to the AUs and I/Os to meet established criteria.

The final step consisted of the resultant estimation of AU and I/O speed requirements to accommodate these loads, allowing for the operations necessary to transfer data on the TLM and Digital Data buses.

9.2.1 Computational Requirements

MSS functional data processing requirements for the command and control central processor have been specified and identified in respect to the computational support requirements for MSS subsystems. An overall summary of the computational load required to meet total subsystem computational needs, with load figures for critical and non-critical subsystem functions is as follows:

¹In the final configuration, each "AU" or "I/O" may actually be a set of similar elements for further redundancy and reliability. For convenience in this report each potential AU set or I/O set will be referred to as an "AU" or "I/O" respectively.

SUBSYSTEM	CRITICAL KEAPS	NON-CRITICAL KEAPS	TOTAL REQUIRED KEAPS
G&C	-	38.0	38.0
ECLSS	2.3	26.2	28.5
EPS	22.6	135.8	158.4
RCS	8.1	2.8	10.9
STRUCTURES	-	0.8	0.8
CREW	-	1.1	1.1
ISS	177.8	225.5	403.3
TOTALS	210.8	430.2	641.0

9.2.2 AU-I/O Pair Loading

Based on this tabulation, and employing the guideline that all critical functions would be duplicated on each side of the CP, an initial allocation can be made of subsystem functions to each "half" of the CP (i.e., each of the two AU-I/O set combinations). Since the remaining OBCO executive and FI routines encompass about one-half of the non-critical total, a decision was made to allocate all non-critical OBCO functions to one AU-I/O pair, with all other non-critical subsystem functions allocated to the other AU-I/O pair. Critical functions would be performed by both pairs simultaneously. The resultant distribution to each AU-I/O combination would then be as follows:

AU-I/O ₁			AU-I/O ₂	
CP EXECUTIVE	- 84.0K		CP EXECUTIVE	- 84.0K
OBCO M&A EXEC	- 17.6	CRITICAL FUNC- TIONS	OBCO M&A EXEC.	- 17.6
OTHER CRIT. FUNCS.	- 109.2		OTHER CRIT.FUNCS.	- 109.2
OBCO (remaining exec. plus FI)	- 212.4		G&C	- 38.0
		NON- CRITICAL FUNC- TIONS	ECLSS	- 26.2
			EPS	- 135.8
			RCS	- 2.8
			STRUCTURES	- 0.8
			CREW	- 1.1
			OTHER ISS	- 13.1
			BG & ON-DEMAND	- negligible
423.2 KEAPS			428.6 KEAPS	



Note that no attempt has been made to allocate segments of the estimated 84K multiprocessor executive amongst components of the CP. Several approaches to a supervisor rationale are possible (master/slave, distributed fixed modules, "floating" modules, etc.), each of which could impact on the final distribution of all critical and non-critical functions. However, multiprocessor supervisor designs are generally chosen to support a given distribution of applications programs; that is, a segmentation of applications software into AU and I/O processors should be performed first, irrespective of executive considerations. Thus, since executive considerations are still under study, initial consideration has been given to the distribution of applications programs.

The AU-I/O pair loading illustrated can be summarized as follows:

- . Approximately equal loads
- . All critical functions in both pairs
- . Non-critical OBCO operations assigned to one pair with all other non-critical operations assigned to the other AU-I/O pair.

9.2.3 AU-I/O Element Loading

Based upon the previous AU-I/O pair loading, the allocation of functions between the AU and I/O elements which constitute an AU-I/O pair is required. Element allocation began with a detailed investigation of the subsystem functional requirements. Employing the previously established criterion that relatively short programs should be handled by the I/O, initial allocations of each of these functional tasks were made to the AU or I/O. This criterion was augmented by additional criteria, as follows:

- a. In general, most critical functions should be performed in the I/O processor to minimize transfer delays and to minimize possible transmission errors on the internal data bus.
- b. Wherever practical, closely related functional tasks and routines should be performed on the same processor to minimize possible searches and transmissions for common subroutines.

The results of this initial allocation of functional segments appears in Table 9-4.

The next step involved the equitable distribution of these functions between the AU and I/O of each AU-I/O pair. As stated previously, attempts were made to achieve a balance between both of these processor "types", so as to employ a single kind of processing element throughout the MSS. The results of this distribution analysis appear in Tables 9-5 and 9-6.

In addition to the AU totals shown in Tables 9-5 and 9-6, it has been assumed that the I/O processor must perform some processing and control for all routines executed by the AU. This overhead (which is separate from that employed by the CP executive) has nominally been taken to be 10% of the AU load.

Table 9-4. Processor Allocations

SUBSYSTEM	FUNCTION	OPERATIONS (KEAPS)	AU	I/O
G&C	Experiment Module Update Shuttle Alignment Terminal Rendezvous Berthing	BG 38.0	X X X X	
ECLSS	Pumpdown & Repressurization* CO ₂ Management* Electrolysis Control* O ₂ Partial Pressure Control* Humidity & Contamination Control Circulation & Temperature Control O ₂ /N ₂ Control Active Thermal Control Humidity & Urine Recovery Control Wash Water Recovery Food Management Special Life Support	BG 1.7 0.1 0.5 BG BG 2.4 0.1 4.7 0.4 OD 18.6	X X	X X X X X X X X X X
EPS	Solar Array Control EPS Operations* Fuel Cell Control* Lighting Control	0.8 150.0** 7.6 BG	 X X	X X X
RCS	Nitrogen Quantity Balance Hydrogen Gas Control Thrust Value Control* Oxygen Gas Control	0.5 1.9 8.1 0.4	 X	X X X
Structures	Berthing	0.8	X	
Crew	Medical Data Acquisition & Analysis	1.1	X	

BG = background (insignificant)

OD = on demand

*critical functions

**critical portion (15K) will be performed on AU, remainder will be performed on both AU and I/O: 100K on AU and 35K on I/O

Table 9-4. (Cont.)

SUBSYSTEM	FUNCTION	OPERATIONS (KEAPS)	AU	I/O
ISS	Internal Communications Control	1.0		X
	External Communications Control*	29.0		X
	Tracking Control	3.5		X
	CMD & Message Generation*	20.3		X
	Displays & Controls*	5.2		X
	Subsystems Operations*	2.5		X
	Planning & Scheduling	BG	X	
	Logistics Inventory Control	2.3	X	
	Information Storage & Retrieval	0.2		X
	Mission Analysis & Assessment	4.8	X	
	Record Management	BG	X	
	3 Remote Terminals	BG		X
	Printer	1.3		X
	OBCO (M&A) - G&C*	0.9		X
	RCS*	3.5		X
	EPS*	0.7		X
	ECLSS*	13.6		X
	EXT COMM*	0.5		X
	OBCO-M&A Exec*	17.6		X
	OBCO (FI, etc.) - EPS	145.0	X	
	ECLSS	15.0	X	
	all other subsystems	32.4		X
	OBCO-FI, etc. Exec.	20.0	X	X

*Critical functions

**10K will be resident in AU, 10K in I/O

Table 9-5. AU-I/O₁ SUBSYSTEM DISTRIBUTION

SUBSYSTEM	AU		I/O	
	CRITICAL	NON-CRITICAL	CRITICAL	NON-CRITICAL
G&C	-	-	-	-
ECLSS	1.7	-	0.6	-
EPS	-	-	22.6	-
RCS	8.1	-	-	-
STRUCTURES	-	-	-	-
CREW	-	-	-	-
ISS	-	170.0	93.8	42.4
TOTALS	9.8	170.0	117.0	42.4
179.8		159.4		

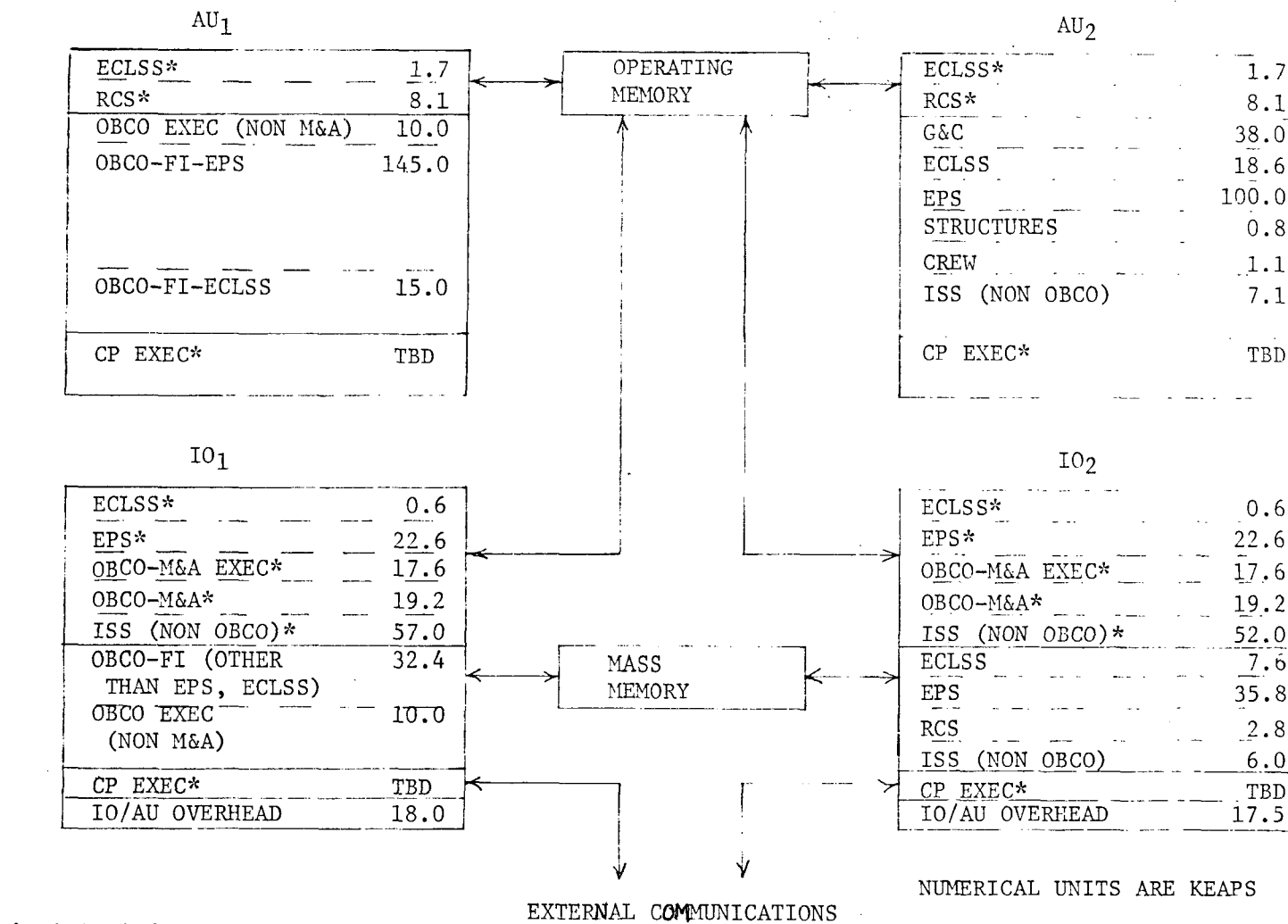
Table 9-6. AU-I/O₂ SUBSYSTEM DISTRIBUTION

SUBSYSTEM	AU		I/O	
	CRITICAL	NON-CRITICAL	CRITICAL	NON-CRITICAL
G&C	-	38.0	-	-
ECLSS	1.7	18.6	0.6	7.6
EPS	-	100.0	22.6	35.8
RCS	8.1	-	-	2.8
STRUCTURES	-	0.8	-	-
CREW	-	1.1	-	-
ISS	-	7.1	93.8	6.0
TOTALS	9.8	165.6	117.0	52.2
175.4		169.2		

Thus, the total expected AU and I/O utilizations are as follows:

AU-I/O ₁ :	Total AU ₁ load	-	179.8 KEAPS
	I/O ₁ applications load	-	159.4
	I/O ₁ overhead (0.10 x 179.8)	-	<u>18.0</u>
	Total I/O ₁ load		177.4 KEAPS
	Total AU ₂ load		357.2 KEAPS
AU-I/O ₂ :	Total AU ₂ load	-	175.4 KEAPS
	I/O ₂ applications load	-	169.2
	I/O ₂ overhead (0.10 x 175.4)	-	<u>17.5</u>
	Total I/O ₂ load		186.7 KEAPS
	Total AU ₂ -I/O ₂ load	-	362.1 KEAPS

Thus, with the functional allocation so indicated, a reasonable balance of the processing loads may be expected. A graphical summary of this resultant allocation is shown in Figure 9-4.



Reproduced from
best available copy.

*critical functions
(a small portion of IO/AU overhead - that
which services the critical 9.8 KEAPS of the
AU - can be considered critical)

Figure 9-4. Detailed CP Subsystem Functional Allocations

10. PROCESSORS PERFORMANCE REQUIREMENTS

The objective of this study was to define the central processor and the preprocessor in terms of their internal organization and required functional and performance characteristics. The preceding sections of this report have presented a number of concepts. This section examines these and presents tradeoff considerations and conclusions regarding their use.

10.1 CENTRAL PROCESSOR TYPE

Two major approaches to the organization of the central processor are as follows: (1) the approach recommended in paragraph 7.1.3 utilized internal logic to detect failures; (2) the approach recommended in paragraph 6.2 places more emphasis on software for detection. The development risks and costs of the two types are felt to be nearly equal. In the one case, the hardware complexity creates the risk and increases the cost. In the other, it is the software complexity.

With regard to the reliability performance, each of the factors of failure coverage, fault isolation and reconfigurability need to be considered. The hardware approach offers greater potential in the ability to detect failures rapidly and prevent error propagation. The fault isolation ability is felt to be equal for the two types since both rely on software and each configuration would have about an equal number of in-flight replaceable units (IFRU's). The one-minute allowance for reconfiguration time (see Table 7-6) means that either approach would probably be acceptable. However, the difficulty in recovering from a propagated error and the necessity to utilize the second computer after the first failure (since cross comparison within and between computers is used for error detection) makes the software approach less attractive.

The physical factors (size, weight, power) are related to the number of hardware components involved. It appears that exclusive of memory and power supply (which should be equivalent for either approach) that the hardware approach involves about 20 percent more for the logic. On the assumption that this is 20 percent of the computer hardware, the net difference between the two is about 4 percent for power, weight and volume.

In view of the low percentage of difference in the physical factors, the equality of development cost and risk, and the greater error detectability, the use of internal logic is preferred.

10.2 CP INTERNAL STRUCTURE

The candidates for the internal structure of the central processor are shown in Figure 10-1 and briefly discussed as follows. The memories, power supplies and internal busing are not shown since these items would be approximately equal for any approach and not major contributors to the selected configuration.

- a. Voters are used upon the address, control and data out of the AUs and IOs. The memory data are encoded for error detection.
- b. The voter of the AUs of the previous type are replaced by comparators. The number of AUs are thus reduced. Both AU sets can perform critical computations and provide continuation of operation upon a first failure.
- c. In this candidate, the comparators are used for both AUs and IOs.
- d. The AUs of this candidate are designed such that an AU set can perform the IO operation. This eliminates the need for voter and IO processor development.

Types a, b, and c are about equal when considering development risk. Type d has an advantage over these since it has less types of IFRUs to be developed. The software development cost of the types b, c, and d is slightly greater than type a since some rollback or reconfiguring software must be provided. However, types a and d represent more hardware development cost (see Table 10-1).

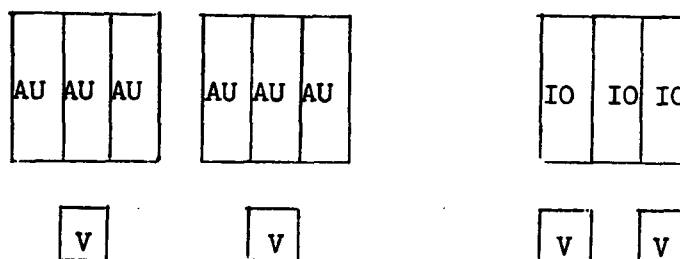
Except for fault isolation, all versions have equal reliability performance. Type a has an advantage with regard to isolation of a faulty element since the voter elements can be used.

As previously noted, the logical portion of the central processor represents about 20 percent of the hardware. Table 10-1 presents the estimation of amount of this logic for each of the candidates.

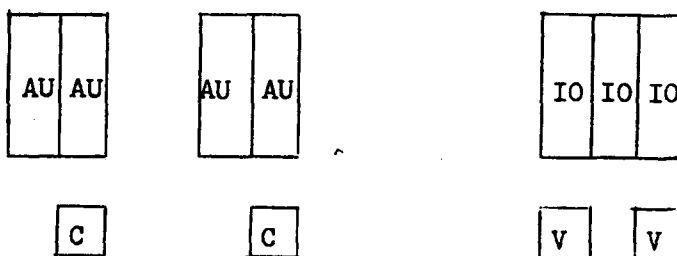
Table 10-1. CP Hardware Comparison

						Selected			
		Type a		Type b		Type c		Type d	
		No.	Equiv. AU *	No.	Equiv. AU *	No.	Equiv. AU *	No.	Equiv. AU *
AU	6	6		4	4	4	4	6	6.6 to 7
IO	3	1 1/2		3	1 1/2	4	2		
V	4	<u>1</u>		2	1/2				
C		<u> </u>		2	<u>1/6</u>	4	<u>1/3</u>	3	<u>.25</u>
Total		8 1/2		6 1/6			6 1/3		6.85 to
Element Type	3			4		3		2	

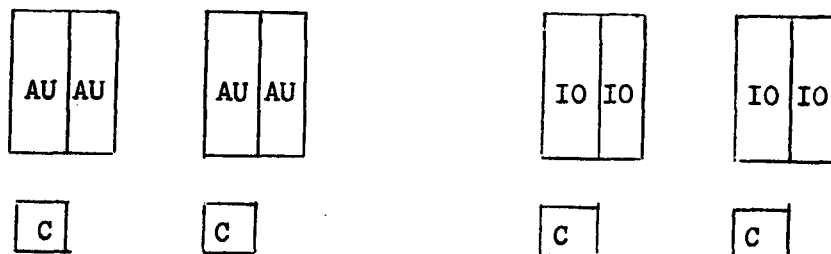
*IO = 1/2 AU, V = 1/4 AU, C = 1/2 AU



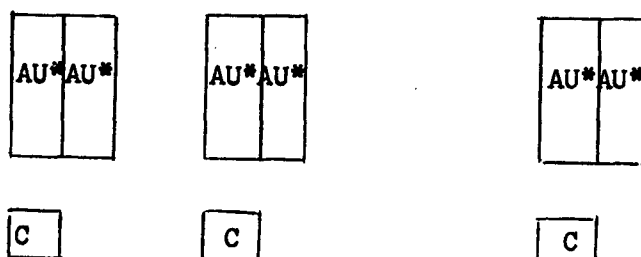
a



b



c



d

LEGEND:

AU = Arithmetic Unit
IO = Input Output Unit
C = Comparator
V = Voter

Figure 10-1. CP Internal Structures

This table indicates that types a and d represent more logic than types b or c. As such, the power, weight, and volume requirements will directly reflect this. The types of elements relate to the IFRU types and indicate favoring type c.

10.3 HOLM VERSUS CONVENTIONAL ARCHITECTURE

10.3.1 Requirements

Section 7.2 presents the features of higher order language machines (HOLM). This section examines the memory savings and cost effectiveness of the HOLM approach. Speed is not considered a key factor since the central processor requirements can be met with equal numbers of AUs and the same type of logic for either conventional or HOLM computers.

Table 10-2 presents a summary of the memory requirements for (a) a conventional architecture, (b) a HOLM where 50-percent memory savings is achieved, and (c) a HOLM similar to that described earlier.

10.3.2 HOLM Implementation Costs

The only quantitative data published at this time relating to the design of a HOLM are presented in the series of articles on the SYMBOL computer. Table 10-3 presents information extracted from these articles and gives an estimate for the HOLM logic.

The arithmetic unit logic attributable to the HOLM features can be determined by considering those features of the SYMBOL machine thus involved. Table 10-3 gives these and shows that an estimated 6850 integrated circuits (IC's) are required in addition to a conventional design of 2500 IC's for implementation of an arithmetic unit and a translator. (For a conventional machine the translation would be done by software - a compiler.)

An evaluation method to determine the cost differential between a HOLM and a conventional design is presented by Kerner and Gillman (see Reference 2 at the end of Section 7.0). Table 10-4 presents this method with numeric values and changes felt to be applicable to the MSS.

The cost differences between the conventional design and HOLM are shown in Table 10-5 for both a 50-percent and a 70-percent memory reduction. These results show that difference can be appreciably greater or less dependent upon which factor of memory reduction is used (50 or 70 percent).

10.3.3 Conclusions of HOLM-Conventional Organization Analysis

The preceding analyses, summarized in Table 10-6, indicate that the total cost can be nearly equal to or several million dollars more expensive when a HOLM is developed rather than a conventional machine for the MSS. The actual value is dependent upon what memory reduction is achievable.

The HOLM has greater risk and credibility due to (1) the uncertainty of its memory improvement, (2) the newness of the concept which affects systems and logic design experience and flexibility.

The result of this analysis is that a conventional organization is preferred for the DPA central processors.

Table 10-2. CP Memory Requirements * - 32 Bit Words

	Conventional		HOLM (50% Reduction)		HOLM (70% Reduction)	
	<u>Total</u>	<u>Each M2</u>	<u>Total</u>	<u>Each M2</u>	<u>Total</u>	<u>Each M2</u>
Critical Instructions (Exec.)	8,800	4,400	8,800	4,400	} 8,040	4,020
Critical Instructions (Operations)	15,454	7,727	7,727	3,863		
Critical Data	5,523	2,762	5,523	2,762	5,523	2,762
Critical Variables	6,523	6,523	6,523	6,523	6,523	6,523
Critical Variables Redundancy	<u>6,523</u>		<u>6,523</u>		<u>6,523</u>	
	42,823	21,412	35,096	17,548	26,609	13,305
Parity 25%	<u>10,705</u>	<u>5,328</u>	<u>8,774</u>	<u>4,387</u>	<u>6,652</u>	<u>3,326</u>
	53,528	26,740	43,870	21,935	33,261	16,631
Uncertainty & Growth Factor	<u>107,056</u>	<u>53,480</u>	<u>87,740</u>	<u>43,870</u>	<u>66,522</u>	<u>33,262</u>
	160,584	80,220	131,610	65,805	99,783	49,893
Overlay	16,384	8,192	8,192	4,096	8,192	4,096
Overlay Parity 25%	<u>4,096</u>	<u>2,048</u>	<u>2,048</u>	<u>1,024</u>	<u>2,048</u>	<u>1,024</u>
	181,064	90,470	141,850	70,925	110,023	55,013

* These requirements differ from those presented in paragraph 7.2.2.2 in that (a) the conventional architecture has the overlay reduced to 16K words from 32K words and is not triplicated for uncertainty and growth (see text); (b) the HOLM is shown which has 50% memory reduction in the operations critical instructions, no reduction in the executive, and an overlay of 8K words; and (c) the HOLM shown here with the 70% reduction is the same except that the overlay is reduced from 16K to 8K and not triplicated.

Table 10-3. HOLM Implementation

<u>SYMBOL Element</u>	<u>Function</u>	<u>% of Logic*</u>	<u>Integrated Circuits</u>	<u>Attributable to HOLM Features (Estimate)</u>
CPU - Instruction Sequencer	Master controller of the CPU	15.3	2,800	1,400
Reference Processor	Manages storage & referencing of all data arrays and structures	11.1	2,000	2,000
Arithmetic Processor	Floating point arithmetic	6.1	1,100	
Format Processor	Alpha-numeric string operations	5.3	950	950
Translator	HOL to Polish object string and name table	14.2	2,500	2,500
		Subtotal	9,350	6,850
Interface Processor	Text editing for interactive communication	7.0	1,200	
Channel Controller	IO control, Buffering, Channel Sequencing	5.6	1,000	
Disc Controller	Disc memory interface	2.8	500	
Memory Reclaimer	Supports MC to make memory space reusable	2.0	350	
Memory Controller (MC)	Allocates memory space on demand, performs address arithmetic and manages associative memory for page	15.3	2,800	
System Supervisor	Task scheduler	15.3	2,800	
		Total	18,000**	6,850

*From Figure 2, W. R. Smith, et al, "SYMBOL" SJCC, AFIPS 1971

**Total number indicated in SYMBOL articles

Table 10-4 Cost Evaluation of HOLM and Conventional Design Applied to MSS

Equations:

- (1) Cost differential between two machines, $C = nM + 2P + 2L$
- (2) Manufacturing and development cost,

$$M = (s \cdot b \cdot r) - (g \cdot c_g \cdot r) - \frac{(g \cdot d \cdot c_c \cdot c_o)}{m}$$
- (3) Power consumption cost savings per computer, $P = (p_m - p_l) c_w$
- (4) Launch weight cost savings per computer,

$$L = [(w_m - w_l) l + (p_m - p_l) w w_l] \quad \text{where } n = \text{number of computers}$$

For MSS = 2 for vehicle
 2 for backup
 2 for program development
 2 for qualification testing
 or $n = 8$

where

Value

S	Memory saved, bits	SOM=operating; SMM=mass
b	Bit cost, \$	Operating=.08, Mass=.004
r	High reliability cost factor	3
g	Number of gates added	IC's X gates/IC X 4AU=109,600
c _g	Cost per gate, \$.1
d	Chips per gate	.003
c _c	Development cost per chip, \$	30K
c _o	Fraction of new chips	.3
w _m	Weight of saved memory, lb.	$3.6 \times 10^{-5} S_{OM}$ or $4 \times 10^{-6} S_{MM}$
w _l	Weight of additional logic, lb.	$2.5 \times 10^{-3} g$
p _m	Power saved due to memory reduction, w	$10^{-4} S_{OM}$ or $10^{-6} S_{MM}$
p _l	Power penalty for added logic, w	.9g
c _w	Cost per watt, \$/w	200
w _w	Weight per watt, LB/w	.16
l	Cost to launch one pound, \$/LB	1000

Table 10-5. Cost Savings Between HOLM and Conventional Organizations -
Exclusive of Development - In Dollars

	50%		70%	
	M2	M3	M2	M3
Memory Saved Bits	40K X 33	243K X 33	70K X 33	447K X 33
P	-170,800	1,600	-151,000	2,900
L	-223,500	40,000	-174,800	73,800
M	-104,000	96,300	134,000	177,000
C		-800K		+1,900K

Table 10-6. Summary of HOLM - Conventional Organization Evaluation

<u>Factor</u>	<u>Item</u>	<u>Selected</u>		
		<u>Conventional</u>	<u>HOLM 50%</u>	<u>HOLM 70%</u>
Cost, \$	Manufacture, power, weight	x	x+.8M	x-1.9M
	Hardware development	y	y+1.8M	y+1.8M
	Compiler development and use	.5 M	--	--
	Max Total	x+y+.5M	x+y+2.6M	x+y-.1M

10.4 SUMMARY OF PROCESSORS CHARACTERISTICS

The central processor is a multiprocessor which possesses the features shown in Table 10-7. As noted, a conventional organization is preferred. A memory hierarchy consisting of buffer memories in the processing elements, modular operating and mass memories is provided. The requirements can be met with two arithmetic and input/output processing sets. Each set contains dual units with capability of comparing memory addressing, controls, and processed results.

The central processor utilizes two operating memories for the main storage functions. These memories are supplied by paging techniques with information from a mass memory. Additional off-line storage is provided by an archive memory. The key features of these are tabulated in Table 10-7.

An arithmetic unit provides one million equivalent adds per second capability. An extensive repertoire, including floating point, is incorporated into the design. Modes of operation include the normal computational and the executive. Privileged instructions only executable in the latter are used. Linkage to the executive mode is by interrupts and special instructions.

All input output functions are controlled by the AU's by means of IO control words and commands from the AU's. Once initiated, IO actions proceed independently of the AU's until completed.

Two transformer rectifier sets are used to convert the primary ac voltages to secondary dc voltages. A redundant power distribution capability is provided internal to the CP. Each set contains power circuitry in active redundancy to be able to use either of the secondary sources.

It was noted earlier that the state of art in smaller aerospace computers is well advanced for the type needed for the preprocessors. The typical characteristics achievable from these are shown in Table 10-8.

The physical values shown in Tables 10-7 and 10-8 are achievable with today's packaging capability. The processors are based upon the use of cased devices on multilayer boards. The mass memory utilizes 2 mil plated wire and power strobing and high-density devices with beam leads, hybrid thin-film, and ceramic substrates.



Table 10-7. Technical Characteristics of the Central Processor

Type:

Multiprocessor, conventional organization, parallel, binary, 16/32 bit data and instruction words.

Operating Memory, M2:

Two required, plated wire, NDRO, each consists of five memory modules of 13K x 33 bits maximum, one parity bit per memory word, one parity word exclusive ORed with block address for every five memory words, echo checking of write operations, one microsecond cycle time with interleaving of the five memory modules, maximum capacity of 18K x 33 bits per each module.

Auxiliary Memories:

Mass Memory - M3, Virtual memory using paging methods, error detection using one parity bit per word and one parity word with address exclusive ORed per every four data words; echo checking of write operations, 2 mil plated wire, NDRO, maximum capability of 1280K x 33 bits, modular design based upon 64K modules.

Archive Memory - Magnetic tape storage with $>5 \times 10^6$ bits per cartridge.

Input-Output:

Two required, each contains dual IO units with comparator AU initiated with self-contained control, solid state buffer memory of nominal 2K x 33 words and 200 nanosecond cycle time, interface with Data Bus Control Unit, Telemetry Bus, and Mass Memory.

Arithmetic Set:

Two required, each contains dual arithmetic units with comparator, solid state buffer memory of nominal 2K x 33 words and 200 nanosecond cycle time 1 million equivalent adds per second per set, fixed and floating point with 100-200 instructions.

Physical Estimates:

	Mass Memory	Archive Memory	Multiprocessor Set
Size, cubic inches	3900	1200	1000
Weight, pounds	180	40	290
Power, watts	15	45	400

Table 10-8. Technical Characteristics of the Preprocessor

<p><u>Type:</u></p> <p>Uniprocessor, parallel-binary, 16 bits data, 16/32 bit instruction words.</p>
<p><u>Memory:</u></p> <p>Capacity - 20K word, 17 bit Plated Wire Storage.</p> <p>One bit of parity per 16 bits.</p> <p>Cycle time - 1 microsecond</p>
<p><u>Input/Output:</u></p> <p>One buffered 16 bit parallel input and output channel.</p> <p>Eight external interrupts.</p>
<p><u>Instruction Repertoire:</u></p> <p>Single and double word addressing.</p> <p>Single word non-addressing.</p> <p>Indexing</p> <p>Indirect addressing.</p>
<p><u>Add Times (Fixed Point):</u></p> <p>Add - 4 microseconds</p> <p>Multiply - 20 microseconds</p> <p>Divide - 40 microseconds</p>
<p><u>Special Features:</u></p> <p>Internal and external interrupts</p> <p>General register file usable as index, base or data register</p>
<p><u>Physical (20K x 17 Bits):</u></p> <p>Size - 400 cubic inches</p> <p>Weight - 15 pounds</p> <p>Power - 50 watts</p>

11. DPA DESIGN REQUIREMENTS SPECIFICATION

11.1 DATA PROCESSING ASSEMBLY GENERAL SPECIFICATION

11.1.1 SCOPE

This document establishes the requirements for the Data Processing Assembly (DPA) of the Modular Space Station (MSS):

11.1.2 REQUIREMENTS

This section establishes the performance and design requirements for the Data Processing Assembly. This section also defines and specifies design constraints and standards necessary to assure compatibility of the DPA with other spacecraft subsystems.

11.1.2.1 ITEM DEFINITION

The Data Processing Assembly consists of the following:

Item	Number
Data Acquisition and Control Subassembly (DACS)	1
Central Timing Unit (CTU)	2
Central Processor (CP)	2
Build Up Data Processor (BUDP)	2

The DACS consists of:

	Initial No.	Final No.
Data Bus Control Unit (DBCUC)	4	4
Remote Acquisition and Control Unit (RACU)	60	(TBD)
Digital Data Bus (DDB)	1	1
Remote Processing Unit (RPU)	5	(TBD)

11.1.2.1.1 Functional Description

The DPA performs as the computational and management center of the Modular Space Station. The DPA shall perform the functions of data acquisition and control, computation, data processing, display processing, multiplex system control, data transfer, subsystem monitoring and control.

11.1.2.1.2 Data Processing Assembly

The DPA is a digital data acquisition and processing system whose elements shall consist of a Data Acquisition and Control Subassembly, Central Timing Units, Central Processors, and Buildup Data Processors. The elements shall

be configured such that the loss of a single element shall not cause the loss of any other element of the DPA. The block diagram of the DPA is shown in Figure 11-1.

11.1.2.1.2.1 DPA Configuration

11.1.2.1.2.1.1 Initial Configuration. The initial DPA configuration shall consist of those DPA elements required to meet or exceed the initial spacecraft requirements. The initial DPA configuration shall be obtained by using those elements from the standard (final) configuration required to meet the initial requirements. Table 11-1 presents the quantity and location of the DPA elements for the initial orbit configuration.

11.1.2.1.2.1.2 Growth to the Standard (Final) Configuration. The standard DPA configuration shall meet the final requirements by inserting modules without any modification to the initial configuration.

11.1.2.1.2.1.3 Identical Elements. Each element shall be identical with all elements within the same classification.

11.1.2.1.2.2 Modularity

The DPA shall be designed such that the addition and deletion of its elements can be made with ease.

11.1.2.1.2.3 Data Acquisition and Control Subassembly

The DACS shall be the subassembly which provides the communication between a number of physically separate locations and equipments. The DACS shall consist of the following elements.

11.1.2.1.2.3.1 Data Bus Control Unit. The DBCU performs as an input/output device for the central processor, controls the information on the data bus, and provides the buffering, formatting, protective coding and checking of the digital data bus data.

11.1.2.1.2.3.2 Digital Data Bus. The DDB provides a redundant communications path between the central processors, remote processing units, and remote acquisition and control units located throughout the modular space station.

11.1.2.1.2.3.3 Remote Acquisition and Control Unit. The RACU is the DACS hardware element that provides the standard interface between the data bus and subsystem functional loops.

11.1.2.1.2.3.4 Remote Processing Unit. The RPUs provide the interface between the data bus and subsystem functional loops together with processing capability.

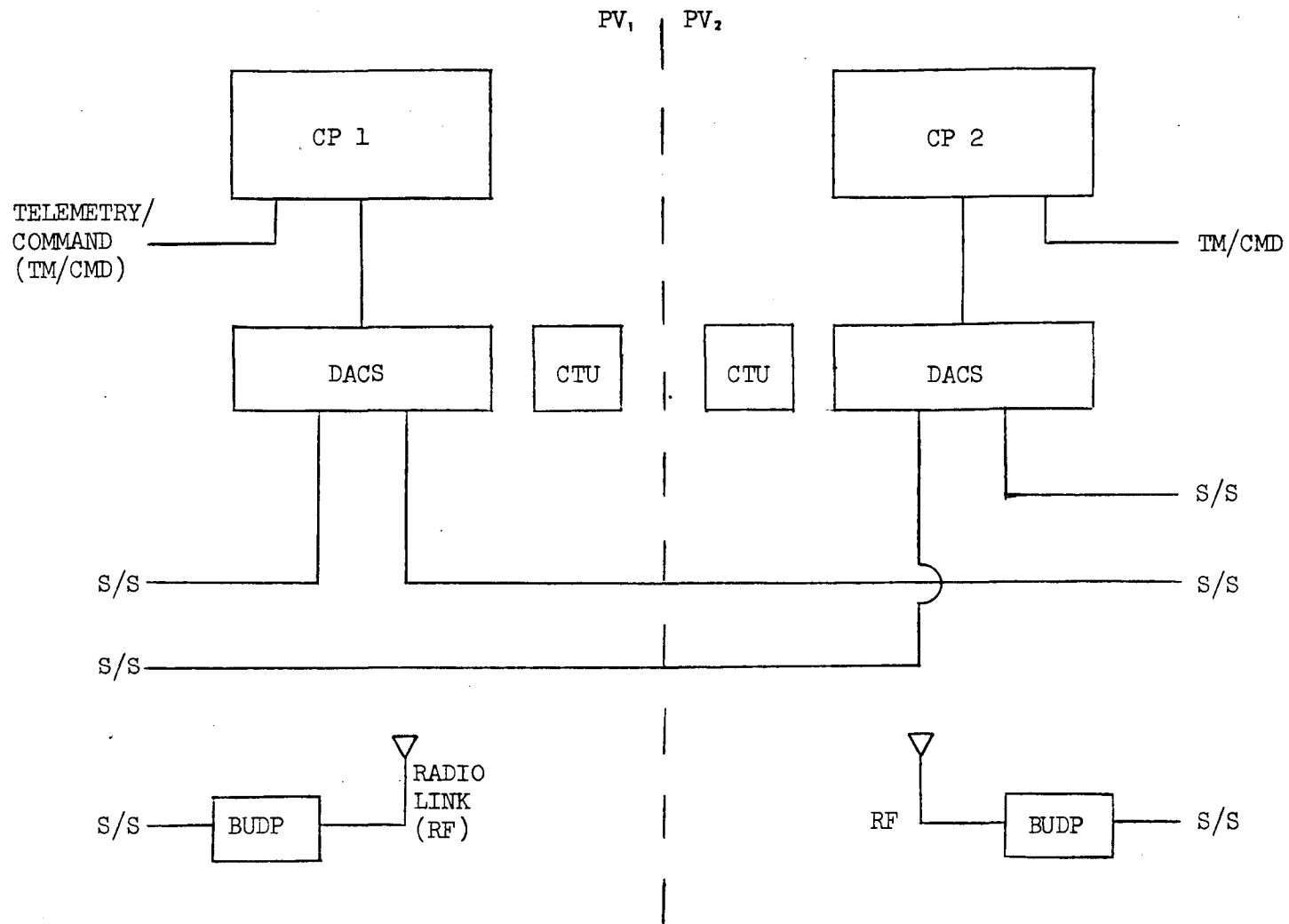


Figure 11-1. Data Processing Assembly

Table 11-1. 6 MAN MSS HARDWARE LOCATION

MODULE DPA SUBASSEMBLY	POWER	CORE	SM1	CARGO 1	SM2	SM3	SM4	QTY
DATA BUS CONTROL			2				2	4
CENTRAL TIMING			2				2	4
CENTRAL PROCESSOR			1				1	2
REMOTE PRO- CESSING UNIT		5						5
REMOTE ACQUIS. & CONT. UNIT	3	12	13	2	8	9	13	60
BUILD-UP DATA PROCESSOR		2						2

11.1.2.1.2.4 Central Timing Unit

The central timing subassembly consists of redundant timing oscillators which provide such functions as vehicle time, timing signals and synchronization pulses.

11.1.2.1.2.5 Central Processor

The Central Processor performs as the computational and management center for the DPA system.

11.1.2.1.2.6 Build Up Data Processor

The BUDP provides the interface between the subsystems in the core and power modules and the radio link to provide command and monitoring capability during station buildup.

11.1.2.2 INTERFACE DEFINITION

11.1.2.2.1 Tolerances

Tolerances shall be as specified herein.

11.1.2.2.2 Electrical Interface

The following requirements for signals, sequences, timing and connections shall apply.

11.1.2.2.2.1 Interfacing Equipment Requirements

Equipment interface shall be as specified in Table 11-2.

11.1.2.2.2.2 Interface with Test Equipment

In-Flight Replacement Units (IFRUs) shall be so designed that when removed from the DPA system and tested at the bench, the total functional test requirement shall be satisfied with GSE and authorized adapters.

11.1.2.2.2.3 Power Supply Interface

Except as modified herein, the equipment shall be designed to comply with requirements in the utilization of electric power and shall provide specified performance when supplied with electric power of three-phase, four-wire "Y" system, having a nominal voltage of 120/208 and a nominal frequency of 400 cycles per second (Hz). Power consumption shall be less than TBD watts.

Table 11-2. Electrical Interfaces (TBD)

SOURCE	DEST.	SIGNAL	TYPE	ELECTRICAL CHARACTERISTICS	PARAMETERS	COMMENTS



11.1.2.2.2.4 Control Panel Interface

The DPA shall be capable of connection to a control panel to provide as a minimum the CP control capabilities specified herein. These control capabilities shall include the following:

- Mode control
- Register access and load
- Memory access and load
- Monitor while running

11.1.2.3 SPECIAL PERFORMANCE REQUIREMENTS

11.1.2.3.1 Data Acquisition and Control Subassembly

The DACS shall provide the capability specified in the following subparagraphs.

11.1.2.3.1.1 Digital Data Bus

11.1.2.3.1.1.1 Data Transmission. Data shall be transmitted over hardwire with a word serial, bit serial time division multiplexed format.

11.1.2.3.1.1.2 Bus Geometry. The bus geometry shall be unidirectional buses for the equipment interfaces. These equipment buses shall be interconnected by a bidirectional central bus.

11.1.2.3.1.1.3 Bus Coupling. Transformer (ac) coupling to the buses shall be provided. The data bus shall use modems to interface with other DACS elements.

11.1.2.3.1.1.4 Bus Configuration. Two dual redundant bus sets with one set in each pressure volume shall be provided. Both sets shall be provided in each of the two spacecraft central control modules.

11.1.2.3.1.1.5 Bus Interface. All four buses shall be connected to each DBCU and provide each CP with redundant paths to all DACS elements.

11.1.2.3.1.1.6 Bus Rate. The bus shall have a data transfer rate of 10 megabits/sec.

11.1.2.3.1.1.7 Bus Data Coding. Error detecting codes shall be used. Manchester coding shall be used to transfer clocking signals.

11.1.2.3.1.2 Remote Acquisition and Control Unit

11.1.2.3.1.2.1 Bus Interface. The RACUs shall accept serial digital data in a standard format from two data bus modems.

11.1.2.3.1.2.2 Error Coding. Error detecting codes shall be generated and checked for transmitted and received bus data.

11.1.2.3.1.2.3 Subsystem Interface. A single subsystem interface with capability of

- a. Input - 128 analog
 - 128 discrete or 96 discrete and 1 digital 32 bit word
- b. Output - 128 discrete or 96 discrete and 1 digital 32 bit word

shall be provided. When inputting or outputting digital words, discretes can be used for word identification.

11.1.2.3.1.2.4 Critical Subsystem Connection. The RACUs shall be connected to a critical subsystem to provide dual independent measurements and commands.

11.1.2.3.1.2.5 Non-critical Subsystem Connection. The RACU(s) shall be connected to non-critical subsystems to provide non-redundant measurements and commands.

11.1.2.3.1.2.6 Data Conversion. Acquired bus or subsystem data shall be converted to generate compatible signals and controls. Data buffering and data transfer control shall be provided.

11.1.2.3.1.2.7 Control. Operation of a RACU shall be under control of information from the CP over the digital bus.

11.1.2.3.1.2.8 Power Connection. The RACU shall be designed to permit power connections to be one of the two following ways:

- a. Single Source - For this connection, the RACUs for critical subsystems shall be connected to opposite secondary sources in a module. Fault isolation may require more manual participation.
- b. Dual Source - For this capability, the RACUs shall be capable of being supplied from either of two sources.

11.1.2.3.1.3 Data Bus Control Unit

11.1.2.3.1.3.1 Interface. The DBCU shall provide a single parallel digital interface with the central processor and a quadruple serial digital interface with the digital bus.

11.1.2.3.1.3.2 Bus Control. The DBCU shall control all messages for the DACS and the communication with the RACUs and RPU's via the data bus.

11.1.2.3.1.3.3 Data Buffer. Data storage for message data sequences shall be provided.

11.1.2.3.1.3.4 Message Generation. Message generation shall be under internal DBCU control using information from the central processor.

11.1.2.3.1.3.5 Error Coding. The DBCU shall perform error encoding and detecting on a word or message basis.



11.1.2.3.1.4 Remote Processing Unit

11.1.2.3.1.4.1 Item Definition. The RPU shall consist of a preprocessor and input/output conversion and interface section of an RACU.

11.1.2.3.1.4.2 Performance. The subsystem and digital bus interfaces shall be equivalent to that of an RACU. The RPU shall perform as an RACU when viewed from these interfaces.

11.1.2.3.2 Central Processor

11.1.2.3.2.1 Item Definition

The central processor consists of the Multiprocessor Set, the Mass Memory and the Archive Memory. Figure 11-2 presents a block diagram.

11.1.2.3.2.2 Interconnection

The CP shall communicate with the other elements of the DPA through the DACS. A parallel bus shall be provided to the mass memory. A redundant serial bus for telemetry and command data shall be provided to the radio link equipment. Discretes shall be provided for communication with the other central processor and control console.

11.1.2.3.2.3 Multiprocessor Mechanization

The multiprocessor shall be configured to use hardware comparators and information coding to achieve the reliability performance required.

11.1.2.3.3 Central Timing Unit

11.1.2.3.2.1 Internal Clock

The CTU shall provide timing and synchronization signals from an internal source. This source shall have TBD characteristics and TBD long term frequency stability.

11.1.2.3.4 Buildup Data Processor

11.1.2.3.4.1 Usage

The BUDP shall provide monitor and control capability during early stages of buildup.

11.1.2.3.4.2 Interfaces

The BUDP shall interface with the radio link and key subsystems with hardwiring.

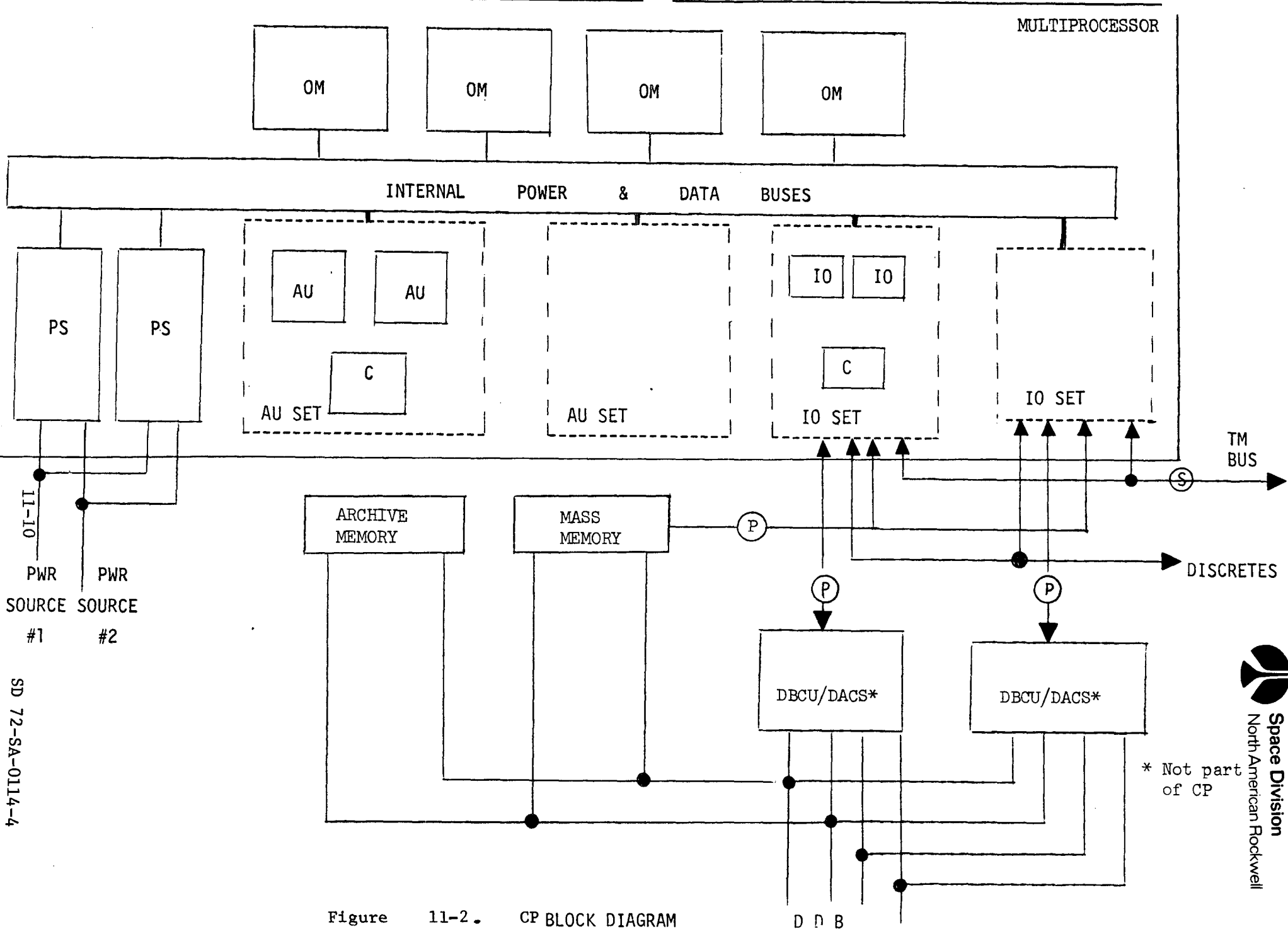


Figure 11-2. CP BLOCK DIAGRAM



11.1.2.4 PHYSICAL CHARACTERISTICS

11.1.2.4.1 Weight

The equipment weight and in-flight replacement units are specified below.
The initial DPA equipment weight shall not exceed the total value specified.

Data Processing Assembly	IFRU Weight	Total
Data Bus Control Unit	(TBD)	(TBD)
Digital Data Bus	(TBD)	(TBD)
Remote Acquisition and Control Unit	(TBD)	(TBD)
Central Timing Unit	18	72
Central Processor		
Arithmetic Unit Set	(TBD)	(TBD)
Input/Output Set	(TBD)	(TBD)
Operating Memory	(TBD)	(TBD)
Power Supply	(TBD)	(TBD)
Mass Memory	(TBD)	(TBD)
Archive Memory	(TBD)	(TBD)
Buildup Data Processor	40	80
	Total	(TBD)

11.1.2.4.2 Size

The limiting boundaries of IFRUs shall be TBD. There shall be no protrusions outside the specified boundaries. Maximum volume of the DPA shall be as specified below.

Data Processing Assembly	Maximum Size (cubic inches)
Data Bus Control Unit	TBD
Digital Data Bus IFRU	TBD
Remote Acquisition and Control Unit	TBD
Central Timing Unit	TBD
Central Processor	TBD
Buildup Data Processor	TBD

11.2 MULTIPROCESSOR SET SPECIFICATION

11.2.1 SCOPE

This document establishes the requirements for the multiprocessor set (MPS) of the central processor for the Data Processing Assembly (DPA) of the Modular Space Station.

11.2.2 REQUIREMENTS

This section establishes the performance and design requirements for the multiprocessor set. This section also defines and specifies design constraints and standards necessary to assure compatibility of the multiprocessor set with other subsystems.

11.2.2.1 Item Definition

The multiprocessor set consists of the following:

<u>IFRU</u>	<u>Number per Multiprocessor</u>
Operating Memory Unit Set	2
Arithmetic Unit Set	2
Input/Output Unit Set	2
Power Supply Unit	2

11.2.2.1.1 Functional Description

The multiprocessor set performs as the computational and management center for the DPA system. The Multiprocessor Set shall perform the functions of computation, data processing, display processing, and multiplex system control and data transfer.

11.2.2.1.2 Multiprocessor Set (MPS)

The Multiprocessor Set is a high-speed, general-purpose digital multiprocessor whose elements shall consist of Operating Memory Unit (M2) set, Arithmetic Unit (AU) set, Input/Output Unit (IOU) set and Power Supply Unit (PSU). The elements shall be configured such that the loss of a single element shall not cause the loss of the entire MPS. The elements shall also be configured such that the loss of a single element shall not cause the loss of any other element of the MPS. The MPS shall be capable of operating with degraded capability when only one M2, one AU set and one IOU set are operative. The block diagram of the computer is shown in Figure 11-3.

11.2.2.1.2.1 Multiprocessor Set Configuration

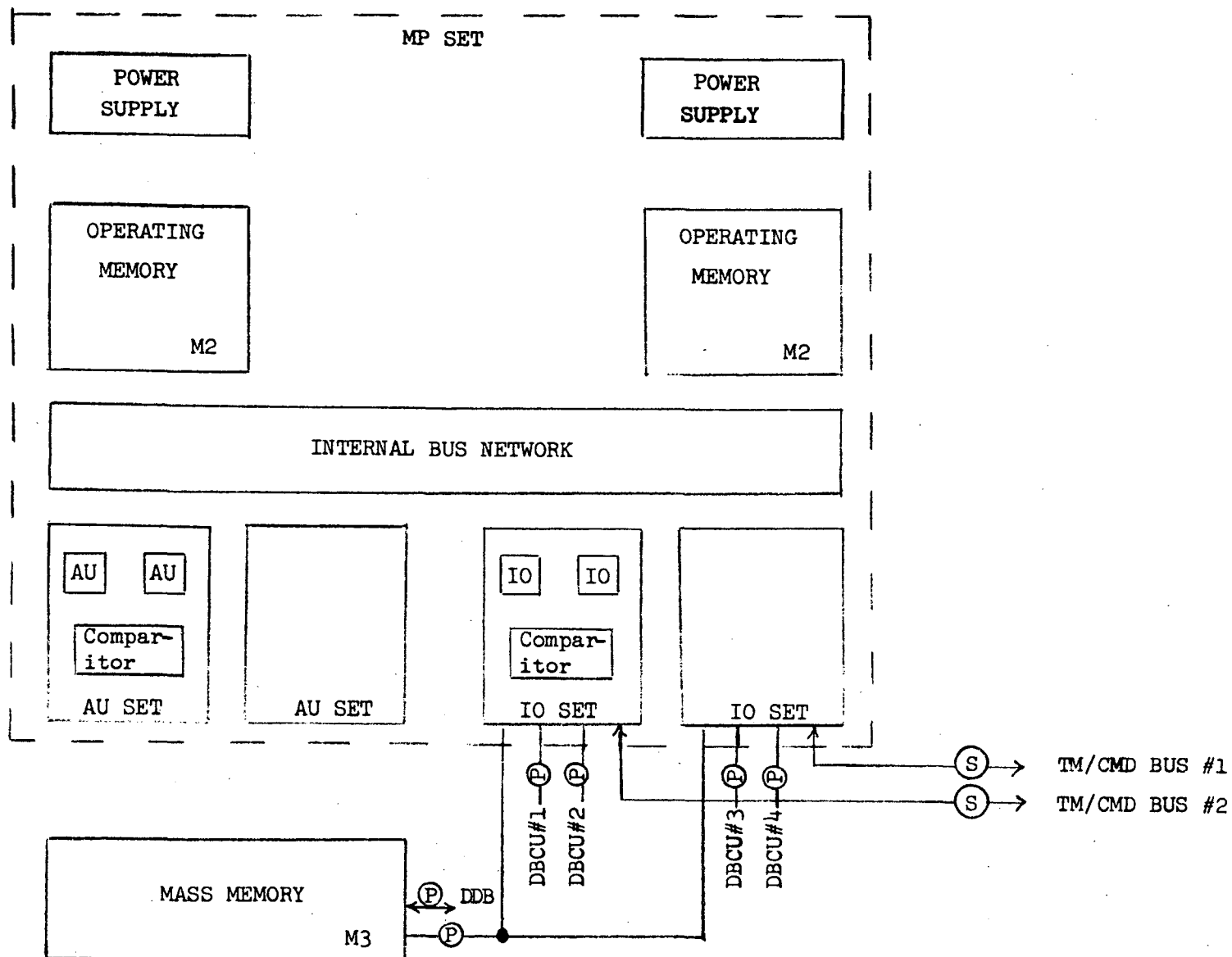


Figure 11-3. Central Processor Block Diagram

- 11.2.2.1.2.1.1 Initial Configuration. The initial multiprocessor configuration shall consist of those multiprocessor units required to meet or exceed the initial multiprocessor sizing requirements.
- 11.2.2.1.2.1.2 Growth to the Final Configuration. The Final multiprocessor configuration shall consist of the initial AU and IO sets plus replacement of larger sized memory units (M2) to meet or exceed the final multiprocessor sizing requirements. Multiprocessor configuration updating to meet final multiprocessor configuration shall be accomplished without any modification to the initial AU and IO sets.
- 11.2.2.1.2.1.3 Identical Submodels. Each unit of each class shall be identical with all units within the same classification type.
- 11.2.2.1.2.2 Modularity. The MPS shall be designed such that the addition and deletion of M2, AU and IOU sets can be made with the insertion or removal of the element, its cables, cooling connection, and its power supply.
- 11.2.2.1.2.3 Arithmetic Unit (AU) Set. The AU set shall contain two AU's and an output comparator. Each AU shall be independent and contain the arithmetic and control logic required to perform the computations and digital data processing. Two dimensional parity checking of AU input data and generation prior to comparison of AU output data shall be provided. Output comparison on all address, control, and data lines between the two AU's in a set shall be made. The AU set shall be capable of initiating and terminating the input-output operation of any IOU set.
- 11.2.2.1.2.4 Input/Output Unit (IOU) Set - The IOU set shall contain two IOU's and an output comparator. Each IOU shall be independent and contain the IO logic to transfer data between the memory units and the peripheral devices. Two dimensional parity checking of IOU input data and generation prior to comparison of IO output data shall be provided. Output comparison on all address, control and data lines between the two IOU's in a set shall be made. The IOU set shall be capable, upon initiation command from an AU set, of continuous data transferral and buffering between the main memory and an input-output channel. The IOU shall have the capability of controlling at least four simultaneous IO channels. IO data flow between the IOU set and memory during simultaneous IO channel operation shall be transferred in parallel words in an interleaved fashion.
- 11.2.2.1.2.5 Operating Memory Unit (M2) Set - The M2 set shall contain five independent plated wire memory modules. A five word data block shall be used to communicate with the AU and IO sets. Four memory modules shall contain program and data and the fifth a parity check word for each block. The M2 set shall contain the read-write circuitry, timing, and access control logic. Error checking using two dimensional parity, address comparison, and write echo checking shall be provided. Virtual memory capability with a mass memory and using paging techniques shall be provided.



11.2.2.1.2.6 Power Supply Unit (PSU) - The power supply unit shall convert the spacecraft primary ac power to the required secondary ac power. The power supply unit shall be capable of supplying full power to the MPS. Dual redundant power distribution shall be provided.

11.2.2.2 Interface Definition

11.2.2.2.1 Tolerances

Tolerances shall be as specified herein.

11.2.2.2.2 Electrical Interface

The following requirements for signals, sequences, timing and connections shall apply.

11.2.2.2.2.1 Interfacing Equipment Requirements. Equipment interface shall be as specified in Table 11-3.

11.2.2.2.2.2 Interface with Test Equipment. In Flight Replacement Units (IFRU's) shall be so designed that when removed from the DPA system and tested at the bench, the total functional test requirement shall be satisfied with GSE and authorized adapters.

11.2.2.2.2.3 Power Supply Interface. Except as modified herein, the equipment shall be designed to comply with requirements in the utilization of electric power and shall provide specified performance when supplied with electric power of three-phase, four-wire "Y" system, having a nominal voltage of 120/208 and a nominal frequency of 400 cycles per second (Hz). Power consumption shall be less than 400 watts.

11.2.2.2.2.4 Control Panel Interface. The computer shall be capable of connection to a control panel via the GSE connector, to provide as a minimum the control capabilities specified herein. These control capabilities shall include the following:

- Mode control
- Register access and load
- Memory access and load
- Monitor while running

11.2.2.3 Special Performance Requirements

11.2.2.3.1 Arithmetic Unit

The Arithmetic Unit shall provide the capability specified in the following subparagraph.

11.2.2.3.1.1 Control Section

Table 11-3. Multiprocesses Interfaces (TBD)

SOURCE	DEST.	SIGNAL	TYPE	ELECTRICAL CHARACTERISTICS	PARAMETERS	COMMENTS



11.2.2.3.1.1.1 Mode Selection. The control section shall provide for selection of the processor state. Two modes shall be provided:

- a. Executive mode. An interrupt or executive call shall cause the AU to enter the executive mode. In this mode the AU shall be capable of executing privileged instructions which cannot be executed in the normal mode.
- b. Normal Mode. Programs operating in the normal mode shall be restricted to the execution of non-privileged instructions. If privileged instructions are attempted to be executed in this mode, an illegal instruction interrupt shall occur.

11.2.2.3.1.1.2 Microprogramming. AU control shall be provided by means of microprogramming.

11.2.2.3.1.2 Time Idle Feature. Each AU shall be capable of performing a Time Idle operation. The operation shall set a value in the timed idle counter. The AU shall perform no more instructions until a timed idle fault or a time idle release. The value shall be automatically decremented until it reaches zero and a time idle fault will be issued. Each AU shall be capable of issuing a timed idle release which shall release the time idle in all other AU's. When a timed idle release is received by an AU with its timed idle counter set, the AU shall proceed to the next instruction and resume normal instruction sequencing.

11.2.2.3.1.3 AU Failure Notification. Each AU shall be capable of issuing an AU failure interrupt to the other AU's if it detects an error in its operations. Each AU shall also contain a fail safe timer which must be reset by the AU at least once per second. If the timer is not reset in the prescribed time, an AU failure interrupt shall be issued to all AU's including the AU whose timer has expired.

11.2.2.3.1.4 Memory Lockout Provision. The MPS shall provide a memory lockout capability instruction. This capability shall enable each AU Set to have sole access to a block of data or instruction in memory. The lockout design shall not depend upon the instruction execution timing of one AU in relation to another AU. The lockout design shall not require more than one privileged instruction execution in setting or releasing the lockout.

11.2.2.3.2 Power-Up Sequencing

When power is applied to the MPS:

- a. All external signals (discrete and I/O commands) shall be in a false state. The IOU's shall be in the idle state.
- b. The AU's shall be activated via the power-on interrupt. Duplication of functions by the various AU's can be prevented by the lockout feature.



The MPS shall be capable of receiving a "Load" signal. When the load signal is received, the activated AU Set shall operate on a read-only program. The read-only program shall have sufficient capability to allow the loading of a program from an external device. Rules a and b above apply to the load procedure as well as power-on.

11.2.2.3.3 Operation Speeds

11.2.2.3.3.1 Arithmetic Unit. The MPS shall be configured in the initial systems with sufficient AU's such that the total throughput for the MPS shall be a capability of 1.9×10^6 operations per second. The operations per second shall be computed as follows:

$$\text{op/sec} = \frac{1}{\sum w_i t_i}$$

where w_i and t_i are defined as follows:

	t_i is the effective time required to perform each operation below	w_i is weighting factor for each operation
1	load	.291
2	store	.254
3	add/subtract*	.125
4	multiply*	.083
5	divide*	.006
6	AND/OR	.032
7	shift (3 bits)	.064
8	branch	.145

*Floating point

Operation types 1 through 6 include an instruction fetch, a full word operand fetch, and a full word operation.

11.2.2.3.4 Arithmetic Unit Features

11.2.2.3.4.1 Word Length. The full word length for processing data shall be 32 bits including the sign bit in the most significant position.

11.2.2.3.4.2 Registers

11.2.2.3.4.2.1 Register File. The AU shall have registers capable of being used as arithmetic registers (accumulators), temporary storage registers and index registers. Each register shall be capable of holding a full computer word.

11.2.2.3.4.2.2 Control Registers. Control registers shall be provided to perform various control functions.



- 11.2. 2.3.4.3 Buffer Memory. A buffer memory (M1) shall provide a fast local storage for each AU. The buffer shall accept five word block data from the operating memory and check for parity. The memory size shall be 2K x 32 words. Associative search control and replacement based upon activity shall be provided.
- 11.2.2.3.4.4 Timing
- 11.2.2.3.4.4.1 Clock Accuracy. The accuracy of the computer clock frequency shall be ± 200 parts per million total including calibration errors, drift due to temperature, short term stability and long term (10 hours) stability.
- 11.2.2.3.4.4.2 Fail-Safe Timer. Each AU shall contain a fail-safe timer which must be reset by the AU (under program control) at least once per second.
- 11.2. 2.3.4.4.3 Real-Time Counter. The equipment shall contain the hardware real-time counter. The counter shall be capable of incrementing or decrementing itself until it reaches a zero state. Upon reaching the zero state, it shall be capable of issuing a system interrupt and automatically reinitializing. The reinitialization value is supplied by the AU and remains constant until a new value is supplied by the AU. The counter shall be capable of being sampled by the AU in such a manner that will not interfere with the register counting function. The resolution of the real-time control shall be equal to or less than one microsecond. The real-time counter shall be incremented or decremented by the computer clock with the accuracy stated in 11.2.2.3.4.4.1. The real-time counter shall be capable of holding a value of one second.
- 11.2.2.3.4.5 Interrupts. A minimum of TBD program interrupts shall be provided. Each interrupt causes the arithmetic unit to take its next instruction from a dedicated memory location associated with the interrupts. Memory addresses that specify the dedicated location shall be permanently assigned. The interrupts if honored shall be taken upon completion of instructions.
- a. Each of the interrupts is individually maskable unless required not to be masked by this specification. Masked interrupts remain pending until unmasked and taken. The unmasking operation shall be structured such that the equipment is capable of handling multiple interrupts without the loss of return address linkages; i.e., an interrupt shall not be taken until the instruction following the unmasking instruction is executed.
 - b. The interrupt commands shall be capable of being configured in a priority structure. The power-on interrupt shall have the highest priority. A higher priority interrupt shall not be inhibited while a lower priority interrupt is being honored. When an interrupt request is honored, it shall be automatically reset. The honored interrupt and all lower or equal priority interrupts shall be automatically masked. They will remain masked until they are unmasked by an unmasking instruction and then a lower priority unmasked pending interrupt shall be honored.



11.2.2.3.4.6 Addressing. The memory accessing instructions shall be capable of addressing all the memory. Capability to address a total of 32,768 full words shall be provided by either direct addressing or base addressing. Capability to address a total of 1,460K full words shall be provided by indirect addressing, paging, or other means.

11.2.2.3.4.6.1 Indexing. The memory accessing instructions shall be indexable. The use of indexing shall not increase the instruction execution time.

11.2.2.3.4.6.2 Indirect Addressing. The memory accessing instructions shall be capable of indirect addressing and/or indexing. The use of indirect address may increase the instruction execution time by one memory cycle per level of indirect addressing.

11.2.2.3.4.7 Instruction Repertoire. The AU shall have a flexible repertoire of instructions. The repertoire shall include the capability for half word, full word floating point arithmetic and fast shift matrix instructions, and a privileged instruction set.

11.2.2.3.5 Operating Memory

11.2.2.3.5.1 Memory Data Protection. The contents of memory shall not be altered because of any conditions of abnormal electric system operation, or due to power supply malfunction, or due to any nominal space station environmental condition.

11.2.2.3.5.2 Memory Word Size. The full word length in memory shall be a minimum of 32 data bits and 1 parity bit.

11.2.2.3.5.3 Memory Organization. The 3 port operating memory shall be organized in 5 equal memory modules. The five word data block shall be placed into these in an interleaved fashion.

11.2.2.3.5.4 Memory Capacity. The MPS shall be initially configured utilizing two operating memory sets so as to provide a total memory capacity of 128K-33 data bit words. The MPS memory capacity shall also be capable of being expanded to 181K-33 data bit words through the replacement of memory modules.

11.2.2.3.5.5 Memory Speed. The effective memory cycle time of each memory module shall be equal to 1.0 microseconds. An operating memory set shall have an effective cycle time of 200 nanoseconds.

11.2.2.3.5.6 Error Detection. Error detection shall be accomplished by

- 1) One parity bit per word
- 2) One vertical parity word per block exclusive ORed with the physical block address
- 3) Echo checks on all write operations

Upon occurrence of an error an interrupt shall be issued to the AU's.

11.2.2.3.5.7 Memory Control

- 11.2.2.3.5.7.1 Memory Access Priority. The memory shall have the capability of being accessed for read or write operations via individual and separate access ports for each user unit in the MPS. Memory access requests from I/O shall be honored first, but no one register shall be allowed to monopolize the memory module.
- 11.2.2.3.5.7.2 Memory Loading and Readout. The memory module shall have the capability to read out the memory and to fill the memory from an external source via the GSE connector as defined below and in Section 11.2.2.2.2.4.
- 11.2.2.3.5.7.2.1 GSE Loading and Readout. The memory load and readout shall be accomplished by means of a GSE connector and shall be independent of the initial contents of memory. The data transfer to and from any memory location shall be in word form.
- 11.2.2.3.5.7.3 Mass Memory. The memory shall provide the control algorithms to transfer data to and from the mass memory via the input out on a page basis.
- 11.2.2.3.5.8 Memory Modularity. The memory shall be of modular design such that all units can be accessed simultaneously. Each module shall contain address and data registers. The minimum size shall be 13K full words, maximum 18K.

11.2.2.3.6 Input/Output Unit Set

The Input/Output Unit Set shall provide the capability for communication with external devices.

- 11.2.2.3.6.1 Input/Output Processors. Each of the two IO processors in a set shall be duplexed into a comparator to check the results of identical processing. These IO processors shall behave similarly to the AU processors with regards to error response and memory interfacing. Each IO processor shall contain a local buffer and use microprogramming control.
- 11.2.2.3.6.2 Data Acquisition System (DACS) Channel. An IOU set shall be capable of communication with a Data Bus Control Unit (DBCUs) on a full word parallel channel. Transfer of data on this channel shall be under control of the IOU using request/acknowledge control signals. This channel shall be capable of transmitting a minimum of 10×10^6 bits per second. External subsystems and assemblies including the computer support set of a mass memory, archive memory, digital printer and control console shall be connected to this IO channel via the remote access and control units, digital data bus, and DBCUs.
- 11.2.2.3.6.3 Telemetry/Command (TM/CMD) Channel. The IOU set shall be capable of communication with the telemetry and command equipment on a bit serial channel at a rate of a minimum of 2.0×10^6 bits per second.



- 11.2.2.3.6.4 Mass Memory Channel. The IOU set shall be capable of communication with the Mass Memory on a full word parallel channel. Transfer of data on this channel shall be under control of the IOU using request/acknowledge control signals.
- 11.2.2.3.6.5 Multiple Channel Operation. The IOU shall be capable of transferring all buffered data on the Data Acquisition System, TM/CMD, OM and MM channels concurrently.
- 11.2.2.3.6.6 Signal Interface Requirements. Cable lengths for IOU to external assemblies shall be a maximum of 15 feet.
- 11.2.2.3.7 Internal Data Bus
- 11.2.2.3.7.1 Structure. The internal data bus shall consist of dedicated 33 bits plus tbd control lines for each AU set and IO set to interconnect to the 3 ports in each operating memory.
- 11.2.2.3.7.2 Bit Rate. The maximum bit rate per wire shall be 5 mbps.
- 11.2.2.4 PHYSICAL CHARACTERISTICS
- 11.2.2.4.1 Weight

The equipment weight and the equipment IFRU are specified below. The equipment weight shall not exceed the total value specified.

	<u>Weight (lbs)</u>
MPS including:	
2 AU's	= 20.0
2 IOU's	= 20.0
2 OMU's	= 250.0
	<hr/>
Including interunit cabling & power supplies	
TOTAL	290.0

11.2.2.4.2 Size

The limiting boundaries of IFRU's shall be TBD. There shall be no protrusions outside the specified boundaries. Maximum volume shall be 6 cubic feet.

11.3 MASS MEMORY SPECIFICATION

11.3.1 SCOPE

This document establishes the requirements for the Mass Memory for the Data Processing Assembly (DPA) of the modular space station.

11.3.2 REQUIREMENTS

This section establishes the performance and design requirements for the Mass Memory.

11.3.2.1 Item Definition

The Mass Memory consists of plated wire memory modules. A memory module is 64K 33-bit words. The initial Mass Memory shall consist of 14 memory modules. Growth to the final configuration of 20 memory modules shall be by inserting modules without any modification to the initial configuration.

11.3.2.1.1 Functional Description

The Mass Memory provides the data and program storage for the multiprocessor set of the DPA.

11.3.2.1.2 Interface Definition

The signal interface between each of the computer's IOs and the Mass Memory consists of a program controlled parallel data channel. In addition to the data channel, automatic load control and error indicating signals are required. The Mass Memory shall also interface with the Digital Data Bus through a Remote Acquisition and Control Unit.

11.3.2.2 Performance Requirements

11.3.2.2.1 Memory Data Protection

The contents of memory shall not be altered because of any condition of abnormal electric system operation or due to power supply malfunction or due to any normal space station environmental condition.

11.3.2.2.2 Memory Word Size

The full word length shall be a minimum of 32 data bits and one parity bit.

11.3.2.2.3 Memory Capacity

The Mass Memory shall be configured using multiple memory units so as to provide a final memory capacity of 1280 33-bit words. Memory units shall be 64K x 33 bit sizes.

11.3.2.2.4 Memory Speed

The effective memory cycle time shall be less than or equal to 10 μ seconds.

11.3.2.2.5 Error Detection

11.3.2.2.5.1 Parity and Address Check. Each data word into/out of the Mass Memory shall be checked for/provided with its parity bit. Each block of four words shall have a check word corresponding to the Exclusive OR of the vertical parity derived from the data and the physical block address and be checked on a data transfer.

11.3.2.2.5.2 Echo Checking for Write Operations. Upon storing a block of data, the block shall be read and verified for correct parity.

11.3.2.2.6 Power Requirements

The Mass Memory shall perform as described herein when supplied 115v 400 cps and shall not consume more than tbd watts.

11.3.2.2.7 Physical Characteristics

11.3.2.2.7.1 Weight. The weight of the Mass Memory shall not exceed tbd pounds.

11.3.2.2.7.2 Size. The volume of the Mass Memory shall be less than tbd cubic feet.

11.4 ARCHIVE MEMORY SPECIFICATION

11.4.1 SCOPE

This document establishes the requirements for the Archive Memory for the Data Processing Assembly (DPA) of the modular space station.

11.4.2 REQUIREMENTS

This section establishes the performance and design requirements for the Archive Memory.

2.1 Item Definition

The equipment is a spaceborne, read and write remote control, cartridge-canister loaded tape storage device.

11.4.2.1.1 Functional Description

The digital tape storage device provides the storage for data and bulk program storage.

11.4.2.1.2 Interface Definition

The signal interface between the computer's IOs and the Archive Memory is provided via the Digital Data Bus using a Remote Acquisition Unit. In addition to the data automatic load control and error indicating signals are interchanged.

11.4.2.2 Performance Requirements

11.4.2.2.1 Tape Buffer and Control Logic

The tape buffer and control logic shall provide the necessary data buffering, data transfer control and transport control. Functions and modes of operation possible shall be

- Read block of data
- Write block of data
- Skip a block forward
- Skip a block backward
- Rewind
- Indicate start point
- Indicate end point

11.4.2.2.2 Tape Transport

The Tape Transport shall have the following capability

Capacity:	$\geq 5 \times 10^6$ bits/cartridge
Tape start:	≤ 25 msec
Tape stop:	≤ 50 msec
Inter record gap:	≤ 4 inches
Read speed:	≤ 120 inches/second
Bit density:	≥ 800 bits/track inch
Data rate:	$\geq 40K$ bits/second

Read operations shall be nondestructive. Tape cartridges shall be removed without use of special tools.

11.4.2.2.3 Power Requirements

The Archive Memory shall perform as described herein when supplied 115v 400 cps and consume not more than 45 watts.

11.4.2.2.4 Physical Characteristics

11.4.2.2.4.1 Weight. The weight of the Archive Memory shall not exceed 40 pounds.

11.4.2.2.4.2 Size. The volume of the Archive Memory shall be less than 1200 cubic inches.

11.5 PREPROCESSOR SPECIFICATIONS

11.5.1 SCOPE

This document establishes the requirements for the Preprocessor for the Data Processing Assembly (DPA) of the Modular Space Station (MSS).

11.5.2 REQUIREMENTS

11.5.2.1 Item Definition

The equipment is a general purpose, stored program, parallel, binary computer.

11.5.2.1.1 Item Diagrams

The functional block diagram of the computer is shown in Figure 11-4.

11.5.2.1.1.1 Functional Description. The computer shall be capable of performing the computational and data processing tasks required of the preprocessor for the DPA application. It shall interface with all necessary equipment and internal devices in order to provide the operational capability required.

11.5.2.1.1.2 Computer Organization. The computer is functionally organized into the following major assemblies.

11.5.2.1.1.2.1 Arithmetic Unit. The Arithmetic Unit (AU) shall be capable of performing logical and arithmetic operations with 16 or 32 bit instruction words. The instruction set shall provide for single and double word addressing, indexing and indirect addressing.

11.5.2.1.1.2.1.1 Data Format. Capability shall be provided for fixed point operation with either data word lengths of 16 or 32 bits, including sign.

11.5.2.1.1.2.1.2 Registers. The AU shall contain sufficient registers to perform the instruction and data fetching and processing operations.

11.5.2.1.1.2.2 Input/Output. The I/O section shall provide for the control of communications between external, internal, or the multiplexing function and the AU or memory section. The I/O shall provide capability to communicate with external peripherals, over a fully buffered parallel I/O channel. (TBD) program interrupts shall be provided.

11.5.2.1.1.2.2.1 Peripheral Channel.

11.5.2.1.1.2.2.1.1 Input Channel Characteristics. Sixteen data input lines shall be provided for the peripheral channel input bus.

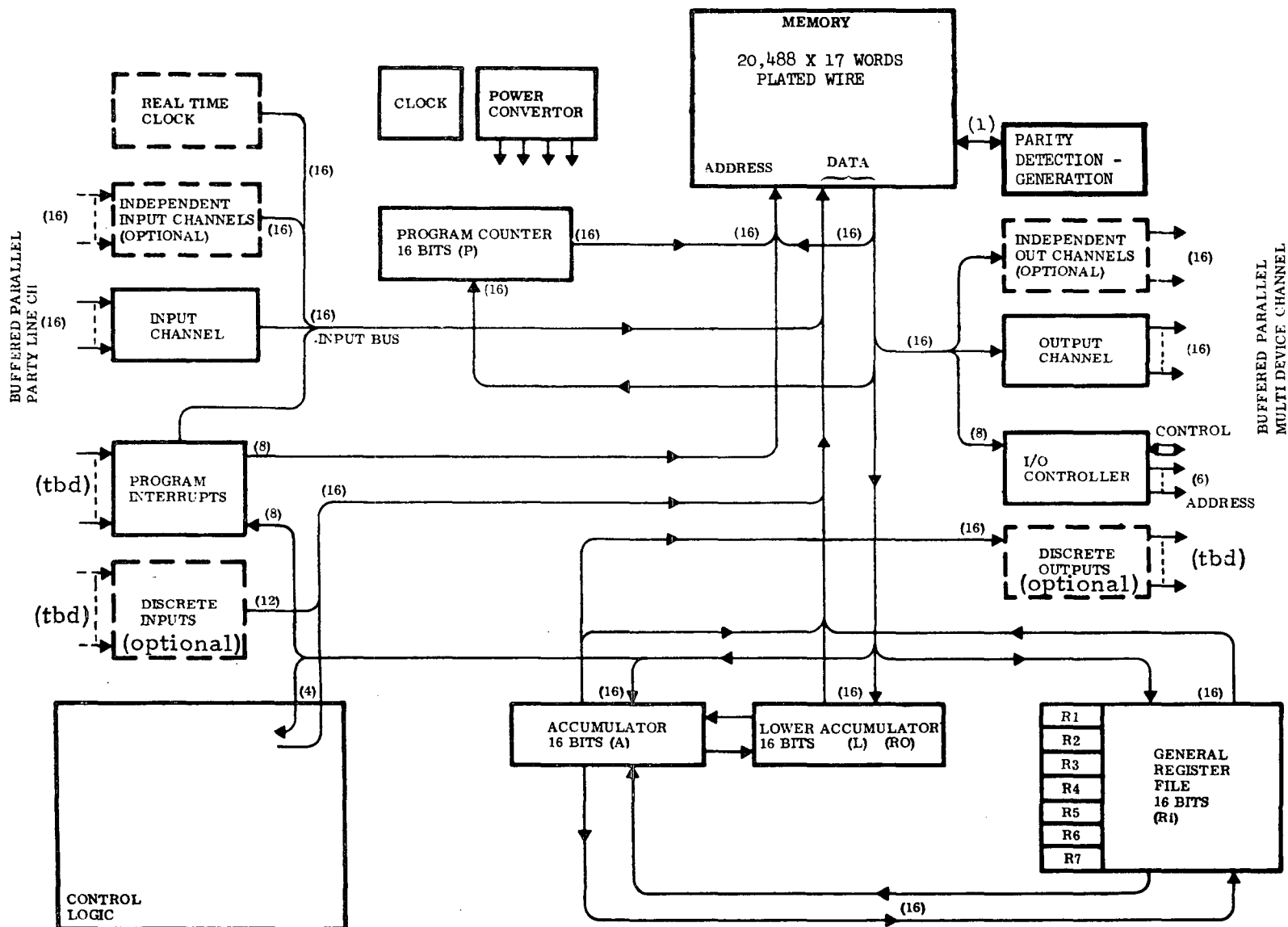


Figure 11-4. Preprocessor Block Diagram

- 11.5.2.1.1.2.2.1.2 Output Channel Characteristics. Sixteen data output lines shall be provided for the data output bus.
- 11.5.2.1.1.2.2.1.3 Addressing and Control. Six address lines and an initiate signal shall be provided for peripheral address and control.
- 11.5.2.1.1.2.2.2 Program Interrupt Channels. (TBD) external program interrupts shall be provided.

A hardware priority shall be provided to resolve simultaneous requests. Each channel shall have a dedicated location in memory associated with it. When a request is to be honored, service to the other channels shall be automatically suspended. The suspension shall remain in effect until an instruction is executed to resume normal operations.

The program interrupt interface shall be fully buffered and shall include an execution acknowledge signal for each interrupt channel.

- 11.5.2.1.1.2.3 Memory Section. The memory system shall be a 20,488 word,(minimum requirement) 17-bit plated wire NDRO memory. Capability shall be provided for coupling to additional memory units for expansion to TBD words in 8K word increments. One memory bit shall be provided for parity checking. An internal interrupt shall be executed upon detection of a parity error.

11.5.2.1.2 Interface Definition

- 11.5.2.1.2.1 Electrical Interfaces. The computer will interface with other equipments of the MSS equipment in accordance with TBD.

11.5.2.2 Characteristics

11.5.2.2.1 Performance

The equipment shall provide satisfactory performance of all the functions specified herein when subjected to the environments or any combination thereof of the Space Station. Satisfactory performance is the performance of a function within its upper and lower limits as specified with the respective functional parameters.

- 11.5.2.2.1.1 Arithmetic Unit. The Arithmetic Unit performance characteristics are summarized in Table 11-4.

- 11.5.2.2.1.2 Memory Section. The Memory Section shall include the following features.

8,192 words of 17 bits
650 nanosec access time
1 μ sec cycle time
Random access
Nondestructive readout
Nonvolatile



Table 11-4. Preprocessor Performance Characteristics

Type	Stored program, parallel general purpose digital computer
Number System	Binary, fixed point, two's complement
Organization	Conventional
Data Word Length	16 bits, including sign
Instruction Word Length	16 bits or 32 bits
Memory Addressing	65,536 words directly addressable - displacement addressing also provided
Memory Speed	1 μ sec cycle time; 650 n sec access time
Basic Clock Rate	1 MHz
Register Complement	Accumulator - 16 bits Lower accumulator - 16 bits Program counter - 16 bits General register file - 7 registers, each 16 bits in length <ul style="list-style-type: none">- indexing- displacement addressing- temporary storage- all are addressable Arithmetic status register
Instruction Repertoire	Instructions of three types exclusive of interrupt and I/O commands <ol style="list-style-type: none">1. Single word addressing2. Double word addressing3. Single word nonaddressing <ul style="list-style-type: none">- Indexing- Indirect addressing- Typical execution times<ul style="list-style-type: none">Add: 4.0 μsecMultiply: 20.0 μsecDivide: 40.0 μsec

11.5.2.2.1.3 Input-Output (I/O). The I/O section performance characteristics are summarized in Table 11-5.

Table 11-5. Input-Output Performance Characteristics

Interrupts	External
	- (TBD) interrupts
	- (TBD) priority levels, uniquely or collectively maskable
	- Interrupt suspension capability
I/O Channels	Internal
	- (TBD) interrupts
	1 buffered 16-bit parallel input/output channel capable of communication with peripheral devices
	Word rate - 80,000 words/sec max.

11.5.2.2.1.4 Power Requirements

11.5.2.2.1.4.1 Power Levels. The computer shall perform as described herein when supplied with 115 volts 400 cps and consume not more than 50 watts.

11.5.2.2.2 Physical Characteristics

11.5.2.2.2.1 Weight. The weight of the computer shall not exceed 15 pounds.

11.5.2.2.2.2 Size. The size of the computer shall not exceed the dimensions of TBD inches in length, TBD inches in height, and TBD inches in depth. Maximum volume shall be 400 cubic inches.

12. DMS PROCESSOR EEM DEVELOPMENT PLAN

12.1 INTRODUCTION

This development and test plan for the DMS EEM processor presents the schedule and identifies the major tasks. This plan is consistent with and requires inputs and support from the space station information management system (IMS) advanced development technology (ADT) program described in SD 71-240, Information Management Advanced Development Technology Extension Study Plan, North American Rockwell, Space Division, October 1, 1971, and summarized in the next section.

The ADT program is structured to include hardware and software studies leading to specifications and breadboard equipment for investigating key aspects of a spacecraft data handling system. These studies are necessary to support the procurement of the EEM processor. The breadboards and software defined in the ADT are required to enable concept evaluation and IMS integration testing.

12.2 ADT MASTER PROGRAM PLAN

The end objective of the IMS advanced development activities is to establish an integrated data management system test bed, whereby the functional and performance characteristics proposed for the MSS information subsystem can be evaluated. The target date for the availability of the DMS test bed IOC at MSC is 1974; with an anticipated MSS launch date of 1983, several years of DMS operation is possible. In this way the evaluation of the validity of design characteristics by means of a low-cost test bed can reduce the total development cost considerably. Furthermore, the test bed, by its use, requires software; software is the most expensive procurement of this subsystem, outweighing all hardware procurement costs. By spreading the software development over 10 years, not only will the spending rate be less, but the total cost will be less. Software costs are high because software must be completed in a short time and is constrained by hardware limitations, and verification and maintenance documentation is necessary. These factors which are present in such programs as Apollo and F-111 can be reduced by an order of magnitude in rate by spreading the time, defining the hardware as a result of software needs, and approaching software development in a breadboard to the same level of documentation that hardware breadboards utilize.

The IMS advanced development master plan is illustrated in Figure 12-1, extracted from NASA planning data. The 1971 IMS advanced development task provided the parts to the left of the boundary on the figure.

The near-goal (1974) objective would be to assemble a prototype data management system to be used to develop automated subsystem's operations, orbiter payload interface requirements and payload data management operations. Figure 12-2 relates the present ADT BB equipments to this objective by a series of



tasks. Figure 12-3 illustrates how these ADT extension (ADTX) tasks contribute to the definition and procurement schedule of an engineering evaluation model (EEM) of the DMS multiprocessor. The six ADTX study areas are:

TASK A - OPERATIONS CONSOLE BREADBOARD

The contractor shall develop a preliminary system specification for an operations console representative of the concept defined for the space station, develop and conduct evaluation tests, develop the software for display generation and control activation, integrate the console with the DACS breadboard, and develop processor software to control and checkout the console.

TASK B - DACS EXTENSION

The DACS extension tasks will take advantage of the DACS breadboards as a development tool. A test and analysis program will be developed around the DACS breadboard alone; also, the DACS breadboard will be used in combination with other breadboards such as the operations console breadboard, the ECLSS breadboard, along with the other breadboards, will be integrated into a subsystem operations breadboard and used in on-board checkout and automatic control evaluations.

TASK C - SOFTWARE EXTENSION

The software extension task has several purposes that include: define a philosophy for control and OBCO for each subsystem, and develop algorithms for each subsystem; develop and investigate the use of higher order languages for real-time programming applications; and develop the actual software programs to operate and evaluate several breadboard subsystems.

TASK D - DATA PROCESSING ASSEMBLY EXTENSION

The purpose of the data processing assembly (DPA) extension task is to continue definition of the DPA with a primary emphasis on the central multiprocessor. This will be accomplished by both analytical studies to reach a DPA mechanization and simulation/evaluation of those mechanizations using IMSIM models. This task culminates in the specification of an engineering evaluation model (EEM) processor that will be used in conjunction with other breadboard subsystems to evaluate the DPA operational concepts.

TASK E - COMMUNICATIONS TERMINAL BREADBOARD EXTENSION

The basic objectives of these tasks is to identify technical guidelines for the MSS communications system development and the functional requirements for this system. This will be provided by a series of integrated tasks that analyze requirements and techniques, develop breadboard hardware, develop evaluation and test plans, and test and evaluate specific portions of the communication system. These tasks will be followed by the evaluation and test plan and the test of an integrated overall MSS communications system.

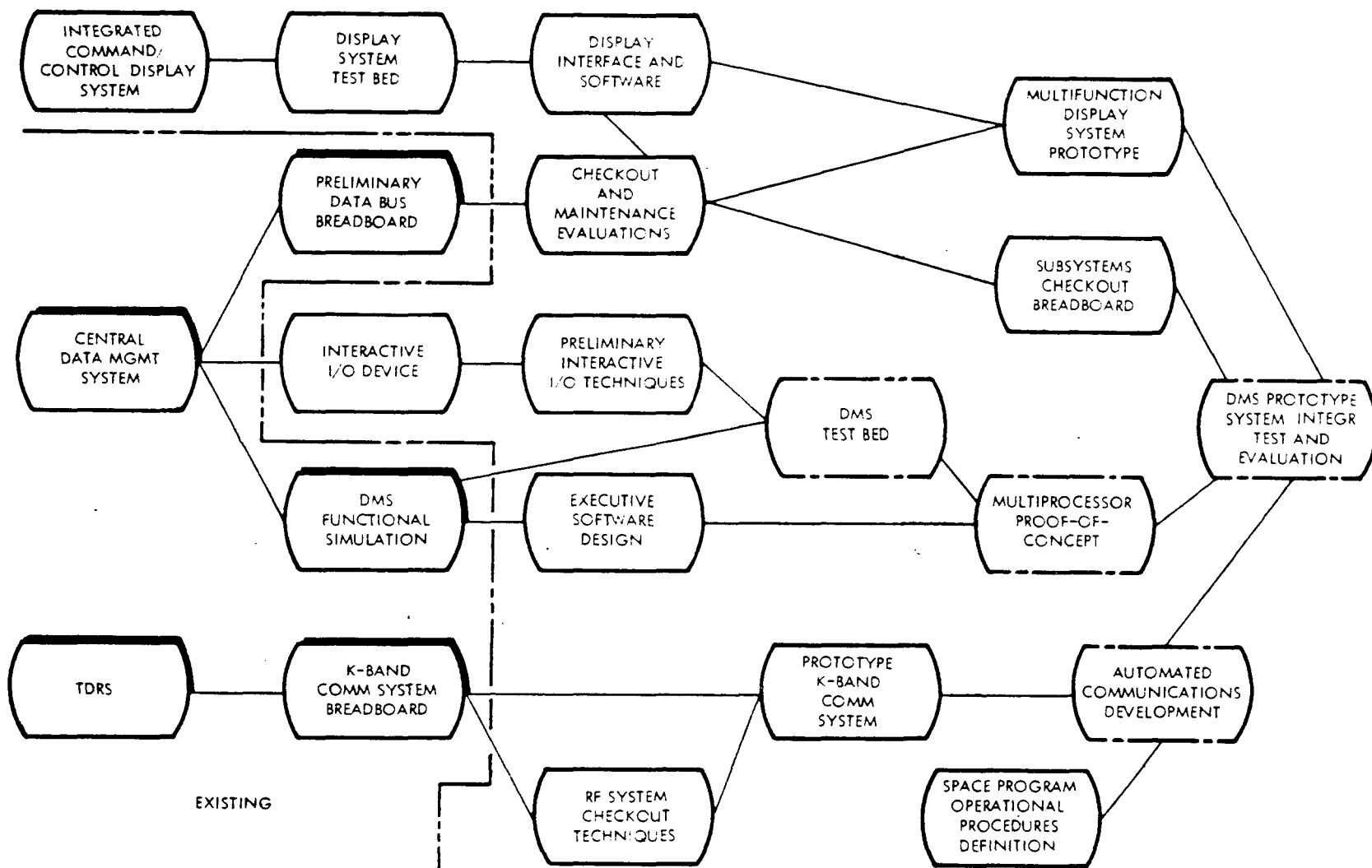


Figure 12-1. Advanced Development Technology Plan

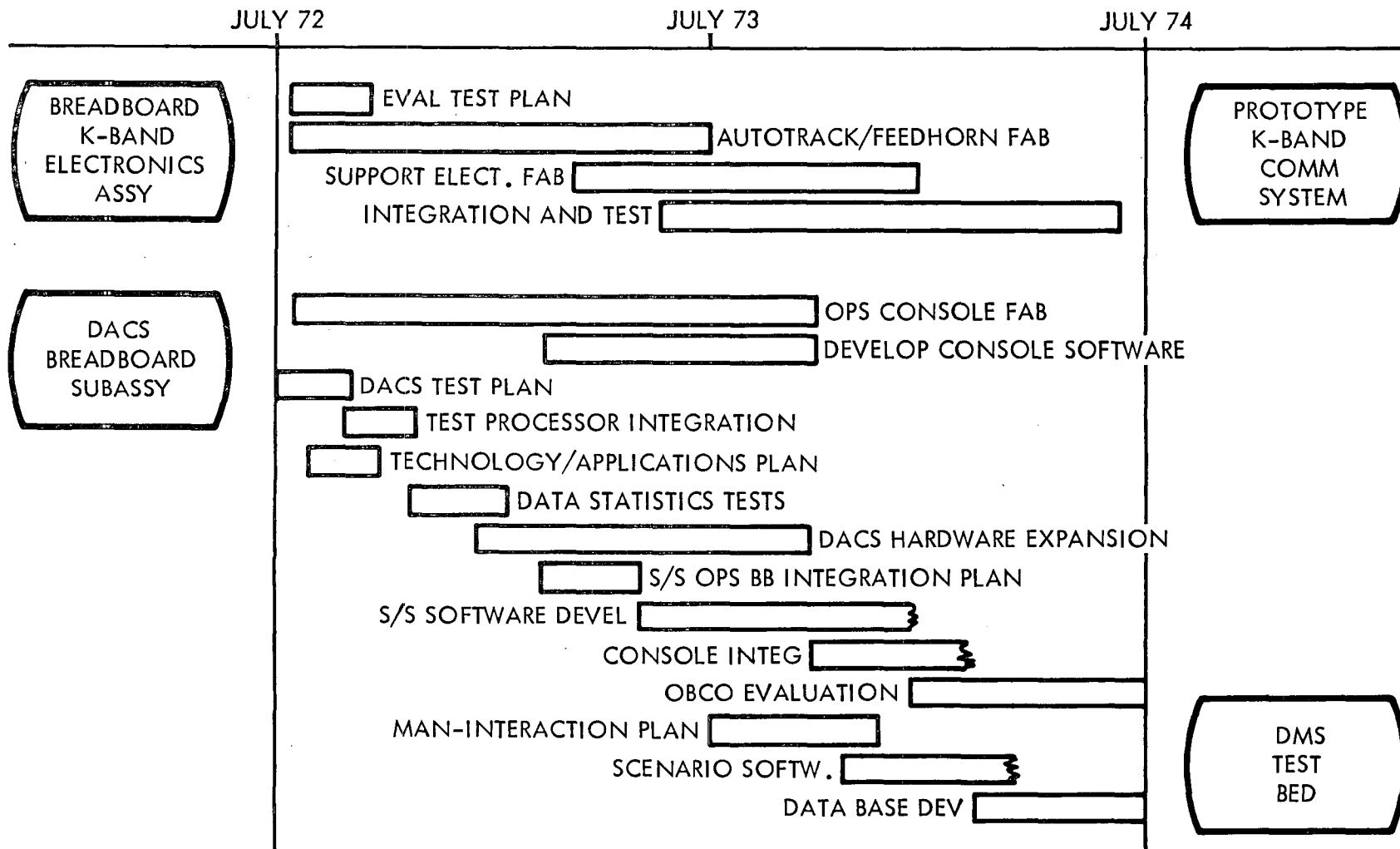


Figure 12-2. ADT Breadboards Related to ADT Plan

FOLDOUT FRAME 1

FOLDOUT FRAME 2

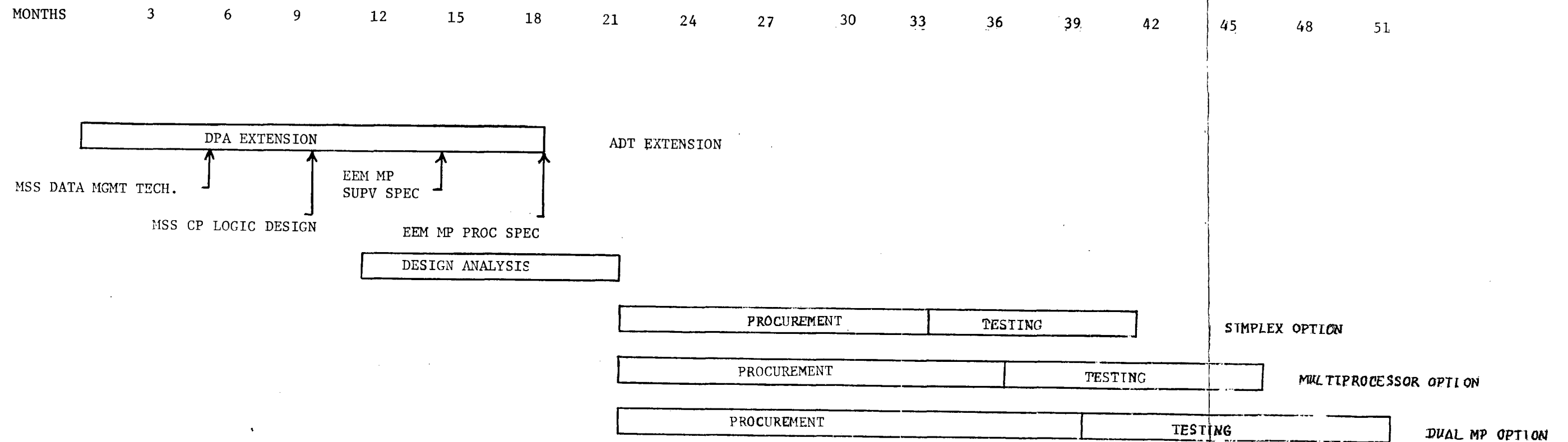


Figure 12-3. Relationship of EEM Processor Development to ADT Extension



TASK F - MAN-MACHINE INTERFACE

The man-machine interface task is intended to accomplish four objectives: (1) define an interactive query language (IQL), which will allow a crew member of optimally communicate with the DPA and requires minimum training or effort; (2) define the display formats that yields optimum machine to man communications; (3) develop a general set of crew procedures that will be used to accomplish various functions that require a combined effort between the DPA and a crew member; and (4) define the DPA software algorithms/logic required to implement the selected IOL and display formats.

12.3 EEM PROCESSOR PLAN

A development program consisting of three phases will provide the EEM processor. Further, dependent upon the level of available funding, various types of processor/breadboards may be procured.

Three possible breadboard processors are:

- a. Simplex Model - 1 AU set; 1 IO set; 1 OM set and usable with the existing DACS breadboard
- b. Multiprocessor Model - 2 AU sets; 2 IO sets; 2 OM sets and usable with the expanded DACS breadboard
- c. Dual MP Model - Duplicate processors each consisting of 2 AU sets; 2 IO sets; 2 OM sets

The three phases and estimated durations are as follows:

Phase I: Requirements Analysis - 10 months

Phase II: Procurement

Simplex Model	- 12 months
Multiprocessor Model	- 15 months
Dual MP Model	- 18 months

Phase III: Testing

Simplex Model	- 8 months (Acceptance)
Multiprocessor Model	- 10 months
Dual MP Model	- 12 months

Figure 12-3 shows the major milestones of the ADT extension study which will provide detailed definition of the central processor requirements and of the EEM processor requirements. This figure shows a requirement of 21 months to define the requirements and perform the EEM processor design analysis. This period would be followed by procurement and testing of the EEM processor - requiring between 20 and 30 months depending on the option chosen.

Figure 12-4 and the following pages describe the major tasks involved in development of the EEM processor.

TASK 1 - REQUIREMENTS ANALYSIS

The major tasks which can be defined directly relating to the definition of the specifications required for procuring the EEM processor are performed during this task. The subtasks are:

ADTX-SUBTASK D1 - EEM MULTIPROCESSOR SUPERVISORY SPECIFICATION

Using the DPA Supervisory Specification previously developed (WBS 85710-2), update and modify this specification to reflect the requirements based upon further evaluation of the DPA configuration. Develop the specification in more detail to cover the requirements unique to the DPA, such as fault detection and reconfiguration. Identify supervisory requirements for the EEM, and prepare a program specification for the EEM executive program.

ADTX-SUBTASK D9 - EEM MULTIPROCESSOR PROCUREMENT SPECIFICATION

Prepare a procurement specification, statement of work, and request for proposal for the EEM multiprocessor. This task is the logical extension WBS 85710-4. It will consolidate the results and recommendations of other tasks of the DPA extension phase into one final document. The performance specifications of the EEM developed under WBS 85710-4 will be developed in more detail to include the internal bus characteristics, functional operation of the memories, control logic, and all system interfaces.

SUBTASK - TEST PLANS

Develop test plans and specifications which include methods of testing and describes test equipment for the evaluation of elements of the processor, the complete processor and the processor when integrated in the DMS.

TASK 2 - EEM PROCESSOR PROCUREMENT

The specification prepared shall be used to obtain a processor for use in the DMS.

The contractor will perform the necessary logic design for the arithmetic, operating memory, and input/output units. The suitable technology shall be selected and the necessary subsystem electronics designed to ensure providing the required performance. Power supplies shall be designed according to the needs.

The necessary hardware fabrication and assembly shall be performed to provide a breadboard system built to best commercial practices.

PHASE 1 - REQUIREMENT ANALYSIS

TASK	TIME - MONTHS
ADTX-D1 EEM Multiprocessor Supervisory Specification	<u>3</u>
ADTX-D9 EEM Multiprocessor Throughput Simulation	<u>4</u>
ADTX-D10 EEM Multiprocessor Procurement Specification	<u>3</u>
Acceptance Test Plan	<u>3</u>
Concept Evaluation Test Plan	<u>3</u>
Integration Plan	<u>3</u>
Support/Inputs:	
ADTX-C Software	
C1 Subsystem Control and OBCO Mechanizations	
C2 Subsystem Control and OBCO Algorithms	
C3 Extension of TOOL for EOSS Applications	
C4 Extension of HAL for EOSS Applications	
C5 Code Generator	
C6 DACS Test Computer Executive Program	
C7 DACS Test Computer Applications Program	
C8 DMS Breadboard Software Development Plan	
C9 MSS Software Configuration Selection	
C10 EEM Software Requirements	
ADTX-D Data Processing Assembly Extension	
D2 DPA Throughput and Authority Analysis	
D3 Central Processor (CP) Logic/Interface Design	
D4 CP Throughput Simulation Plan	
D5 CP Throughput Simulation	
D6 CP Internal Bus Mechanization	
D7 CP Data Management Techniques	
D8 CP Configuration Selection	

Figure 12-4. EEM Processor Development Schedule (Sheet 1 of 4)

PHASE 2 - PROCUREMENT

TASK	TIME - MONTHS
EEM Processor	12 to 18
Software Development	
Support/Inputs:	
Phase 1 tasks	

Figure 12-4. EEM Processor Development Schedule (Sheet 2 of 4)

PHASE 3 - TESTS

TASK	TIME - MONTHS
Acceptance Tasks Construction Design Characteristics Operability Support Inputs: Phases 1 and 2	2 to 3
Concept Evaluation Tests Key Features DPA Performance Failure Tolerance MSS Application Support/Inputs: Phases 1 and 2 ADTX-A Operations Console Breadboard A1 Operations Console System Specification A2 Operations Console Fabrication/Assembly A3 DACS Integration Plan A4 Conduct Acceptance and Integration Tests A5 IBM 360 Operation Console Program ADTX-B DACS Extension B1 Technology Test Plan B2 Application Test Plan B3 Technology/Application Tests B4 Maintenance Evaluation B5 Statistical Model Analysis B6 Statistical Data Tests B7 Statistical Data Analysis B8 Subsystem Operations Breadboard Integration Plan B9 DACS Hardware Extension B10 Subsystem Breadboards Interface Modification B11 Subsystems Operations Breadboard Integration B12 DACS Usage Guidelines and Interface Control Documents	4 to 6

Figure 12-4. EEM Processor Development Schedule (Sheet 3 of 4)

C72-533/201

PHASE 3 - TESTS

TASK	TIME - MONTHS
Integration Tests Hardware Interface Verification Hardware-Software Compatibility Functional Performance	2 to 3
Support/Inputs: Phases 1, 2, ADTXA, B ADTX-F Man Machine Interface F1 Interactive Query Language (IQL) and Display Guidelines F2 Man-Machine Operations Analysis F3 Candidate IQL Definition F4 Candidate Controls and Displays Definition F5 Candidate Crew Procedures Definition F6 Test and Development Plans F7 Controls and Displays Simulation Software F8 Evaluation Scenarios F9 Evaluation Scenario Software F10 Man-Machine Interactive Evaluations F11 Interactive Query Language Specification F12 Controls and Displays Requirements F13 Crew Procedures Guidelines F14 MSS Application Software Requirements F15 MSS Data Base Organization Requirements	
* Length is dependent upon type of EEM processor breadboard funded.	

Figure 12-4. EEM Processor Development Schedule (Sheet 4 of 4)

C72-533/201

TASK 3 - SOFTWARE DEVELOPMENT AND PROCUREMENT

The contractor will develop the software programs required to operate and self-test the processor. This shall include support software such as an assembler and program loader in addition to the executive and selected bread-board applications program according to specifications prepared by ADTX tasks.

TASK 4 - ACCEPTANCE TEST

The EEM processor's capability shall be demonstrated according to the specification and acceptance test plan to the designated NASA representatives at the contractor's facility prior to delivery.

The elements of the processor shall be tested individually and when integrated as a computer. The test data shall be collected and analyzed with respect to performance required by the specification. Tests shall be conducted according to the test plans. Developed software shall be tested and validated.

The acceptance tests shall verify the processor performance, design characteristics, construction, and operability. Methods of verification shall include inspection of hardware, review and acceptance of analytical data, demonstrations, tests, and review of test data.

The actions and tests to support the design and development shall be accomplished through, but not limited to, the following:

1. Inspection

- Workmanship
- Documentation

2. Analyses

- Electrical Interface
- Personnel Safety
- Maintainability

3. Demonstrations

- Interface Compatability
- Failure Detection

4. Tests

- Power Interface and Interrupts
- AU, IO, and Operating Memory characteristics as delineated
in the EEM Processor Specification
- Control Panel
- Support Software
- Failure Response including Fail-Safe

TASK 5 - CONCEPT EVALUATION TESTS

The concept evaluation tests shall permit the key features of the EEM multiprocessor to be investigated. These shall include the following:

1. Error Detection
 - Comparator usage
 - Two dimensional parity
 - Write echo
 - Memory protect
 - M2 organization
2. Cache Memory
3. Memory Management with Use of Paging
4. Multiprocessing
5. Reconfigurability

In addition to checking these features shown, the breadboard processor can provide control and testing of the DMS. The following items can be investigated:

1. Performance
 - a. Operability with the DACS and other breadboards
 - b. Transient response in buildup, power up/down, and reconfiguration
 - c. Throughput as to the affect of message structures, processing overhead, delays, queuing, etc.
2. Failure Tolerance
 - a. Redundance with regard to effectiveness
 - b. Error detection and reconfiguration capabilities
3. Application
 - a. Expandability of concepts and flexibility to changes
 - b. Executive functions of scheduling, control and resource allocation
 - c. Support and operational software requirements and implementation



TASK 6 - INTEGRATION TESTS

The processor shall be integrated in the DMS at NASA-Houston and a series of integration tests utilizing the Integration Test Plan and Specification and software developed during Phase I and ADTX, respectively.

The processor, in conjunction with the other DMS elements, is intended to demonstrate the following technology goals:

Failure detection and isolation abilities and techniques of a spacecraft data handling system

Traffic control methods

Aspects of various allocations of operational functions

Performance of elements of a data acquisition and control system

The integration tests for the EEM processor can be classified as follows:

1. Hardware Interface Verification - these tests shall verify the capability of the processor to be utilized with the DMS hardware
2. Hardware - Software Compatibility - the ability of the EEM processor to control and operate the DMS shall be thus demonstrated
3. Functional Performance Verification - these tests shall constitute the DMS integration tests and demonstrate the ability of the processor to perform the above goals. Some analysis of the data acquired may be necessary to support the verification

12.4 SCHEDULING OPTIONS

Figure 12-3 showed the time requirements associated with procuring and testing (1) a simplex model, (2) a multiprocessor model or (3) a dual multiprocessor model. In any event, it will be necessary to procure and evaluate a dual multiprocessor model prior to the procurement of the MSS central processors. There are a number of ways to accomplish this goal ranging all the way from sequential procurement of sufficiently many simplex models to all-out procurement of a dual multiprocessor in one effort.

To illustrate some of the effects of these possible procurement plans, we begin by assuming:

1. MSS IOC at the end of 1984
2. 18 months for MSS subsystems integration

3. 6 months for flight article dual multiprocessor acceptance and qualification testing
4. 18 months for procurement of the MSS flight article dual multiprocessor
5. At least 12 months of evaluation testing with the EEM dual multiprocessor prior to procurement of the flight article dual multiprocessor

Based on these, we can schedule IOC of the EEM dual multiprocessor for mid-1980. These assumptions and three procurement options are illustrated in Figure 12-5.

The first option represents sequential procurement of two simplex models (together forming a multiprocessor) followed by procurement of a second multiprocessor. This option will require the initiation of the requirements definition at the beginning of 1974. This option may also be characterized as having the least risk and by having the lowest peak of funding spread over 7.5 years.

The second option represents sequential procurement of two multiprocessors. Initiation of requirements definition could be delayed a year to the beginning of 1975, but this option would require higher funding peaks and higher risk.

The third option represents procurement of a dual multiprocessor saving another 15 months but requiring the highest funding peak and the highest risk of all the options.

FOLDOUT FRAME 1

FOLDOUT FRAME 2

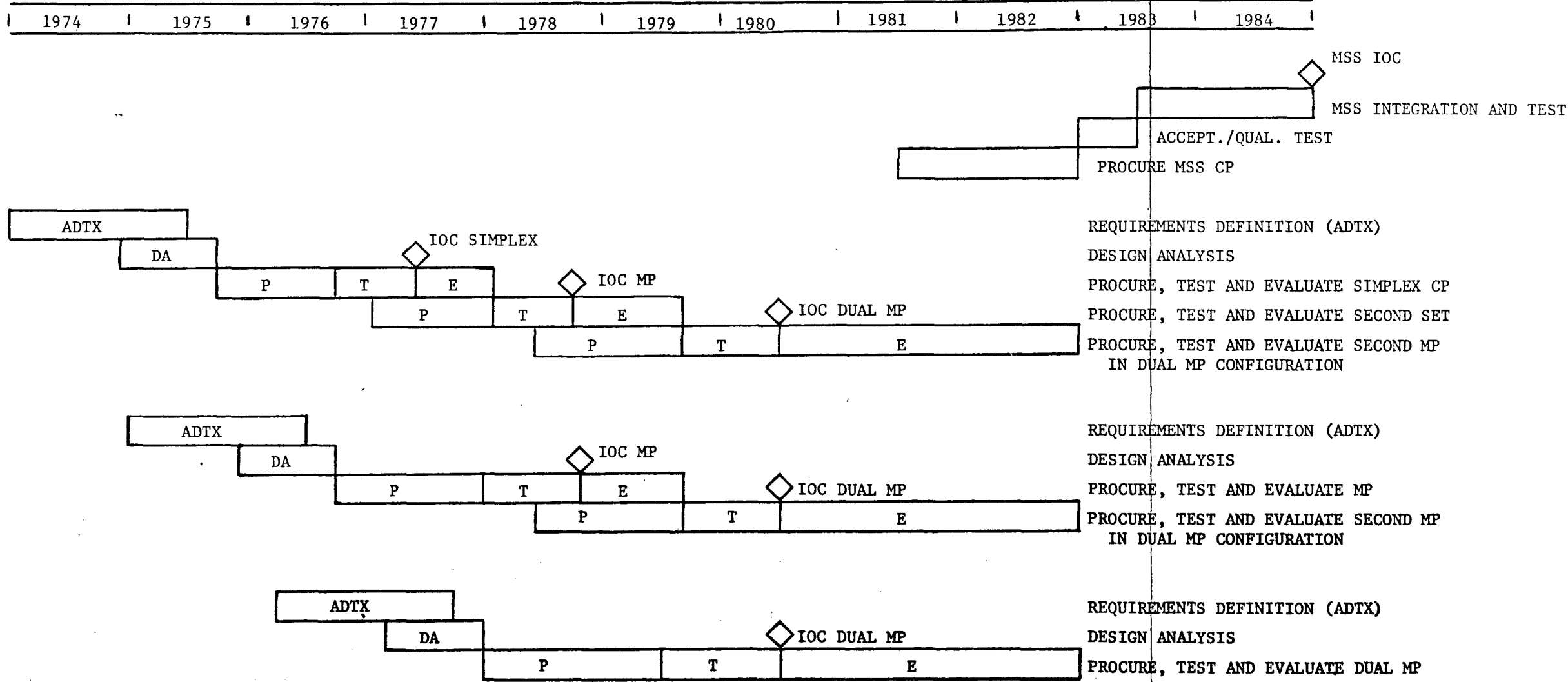


Figure 12-5. Procurement Options for a Dual Multiprocessor EEM